



Własna mini-wtyczka, czyli...

# Akcje w QGIS

# Kim jestem? Co robię? Z jakich narzędzi korzystam?



- od ponad 10 lat zajmuję się GIS-em
- ukończyłem studia geodezyjne oraz podyplomowe informatyczne i data science
- pracuję w Krajowym Ośrodku Wsparcia Rolnictwa OT w Olsztynie
  - na co dzień wykorzystuję Pythona, QGIS, bazy PostgreSQL z PostGIS, Oracle, MySQL oraz sieciowe oraz lokalne plikowe źródła danych
  - większość analiz przestrzennych wykonuję poza QGIS, szczególnie w PostGIS lub Pythonie
  - stworzyłem wewnętrzny geoportal KOWR, którym już od 8 lat administruję
  - zajmuję się integracją danych z wielu źródeł oraz automatyzacją powtarzalnych zadań
- okazjonalnie prowadzę szkolenia z wykorzystania Pythona w QGIS jako trener Enviro Solutions
- w "erze przedojcowskiej" przetłumaczyłem sporą część interfejsu QGIS na polski



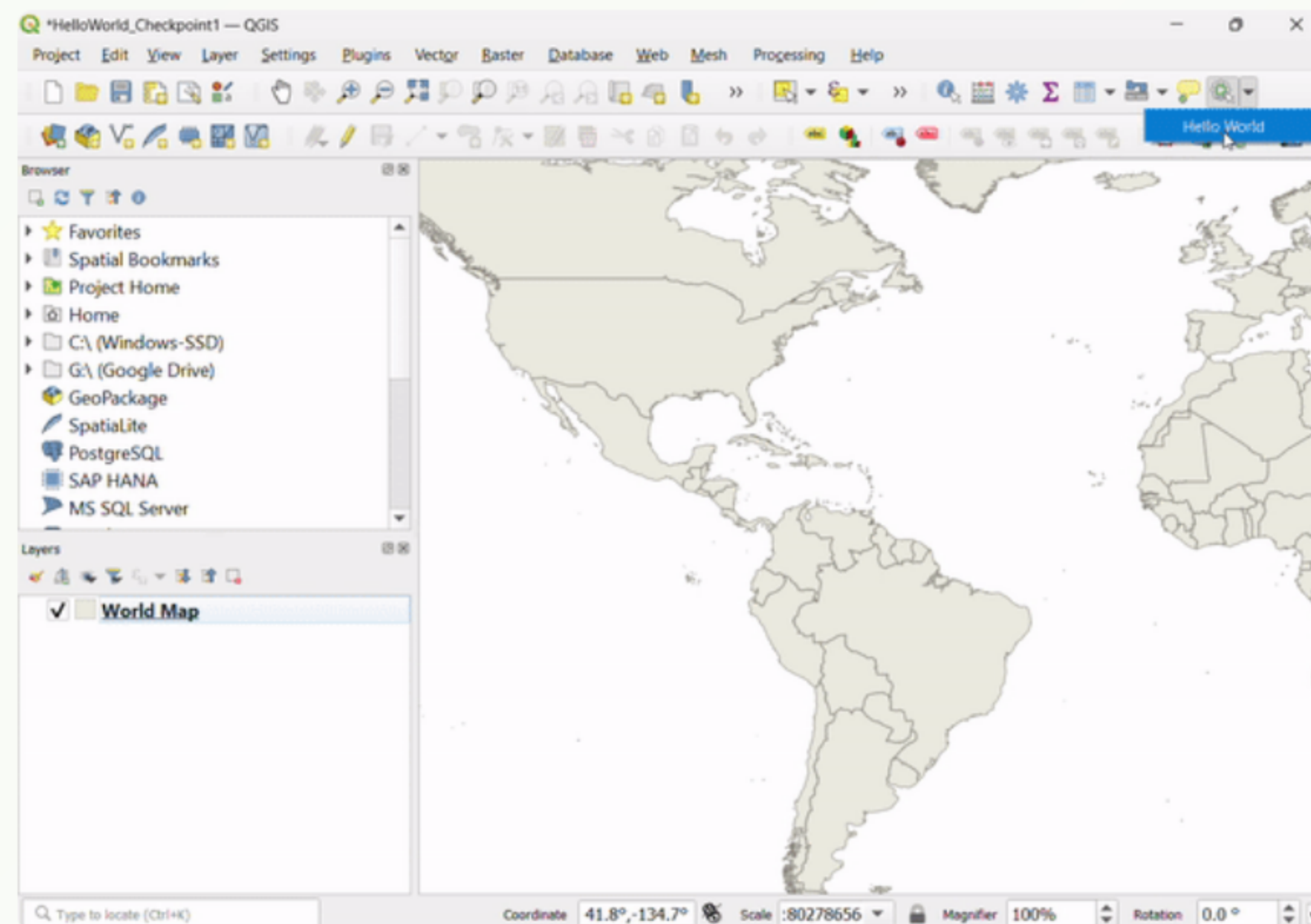
## Czym są akcje w QGIS?

- **Akcje** to wbudowana w QGIS funkcjonalność umożliwiająca wykonanie na obiektach przestrzennych określonych przez użytkownika operacji
- Mogą być uruchamiane poprzez kliknięcie na obiekt (lub wybór z menu kontekstowego)
- Pozwalają na integrację QGIS z zewnętrznymi aplikacjami i usługami
- Zwiększają funkcjonalność programu bez konieczności pisania wtyczek

## Kiedy skorzystać z akcji?

Skorzystaj z akcji, jeśli chcesz dodać do QGIS funkcjonalność, która ma być dostępna dla użytkownika jako interakcja z warstwą.

# Przykłady akcji w QGIS

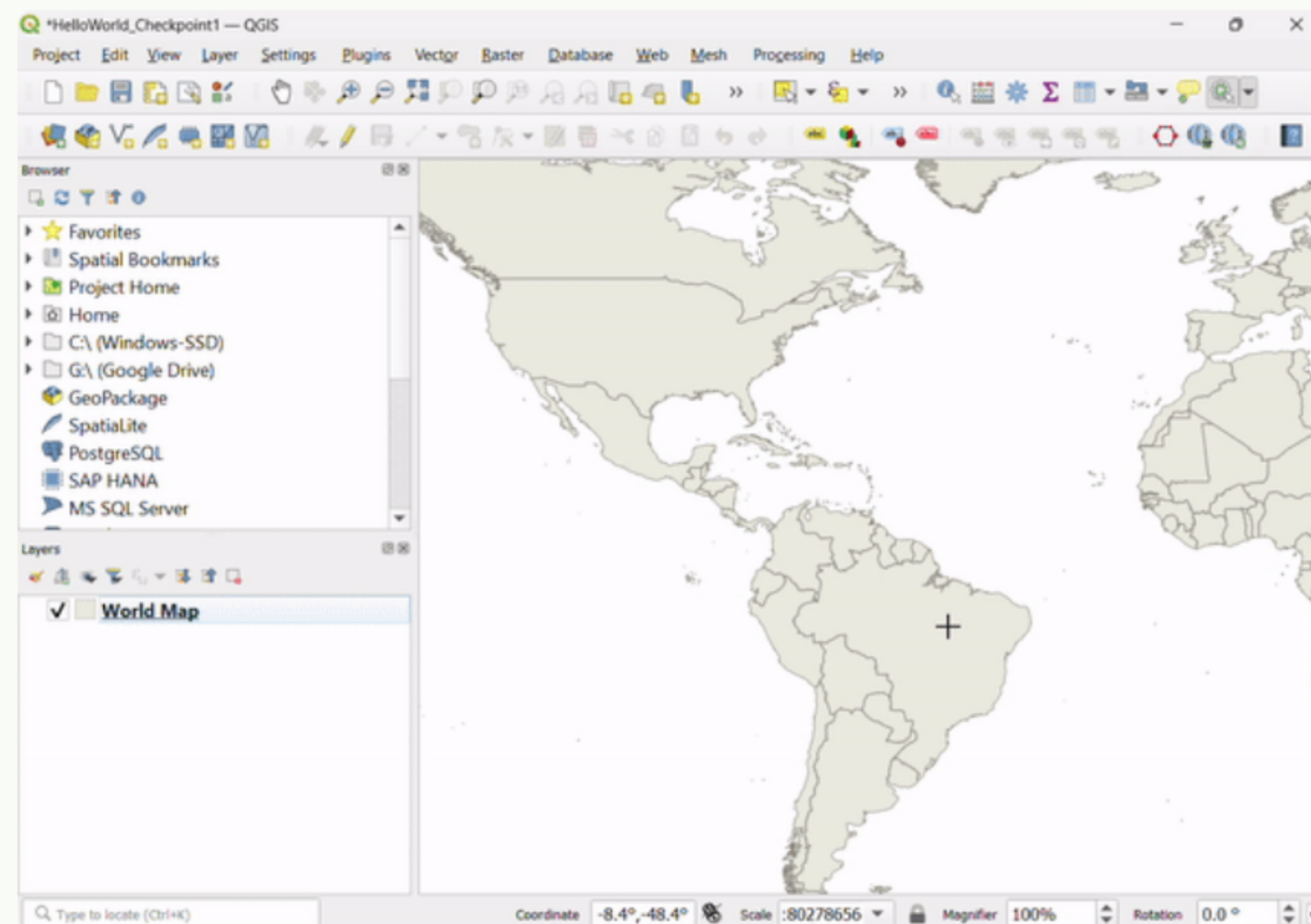


[źródło animacji: SpatialThoughts]

3 / 27



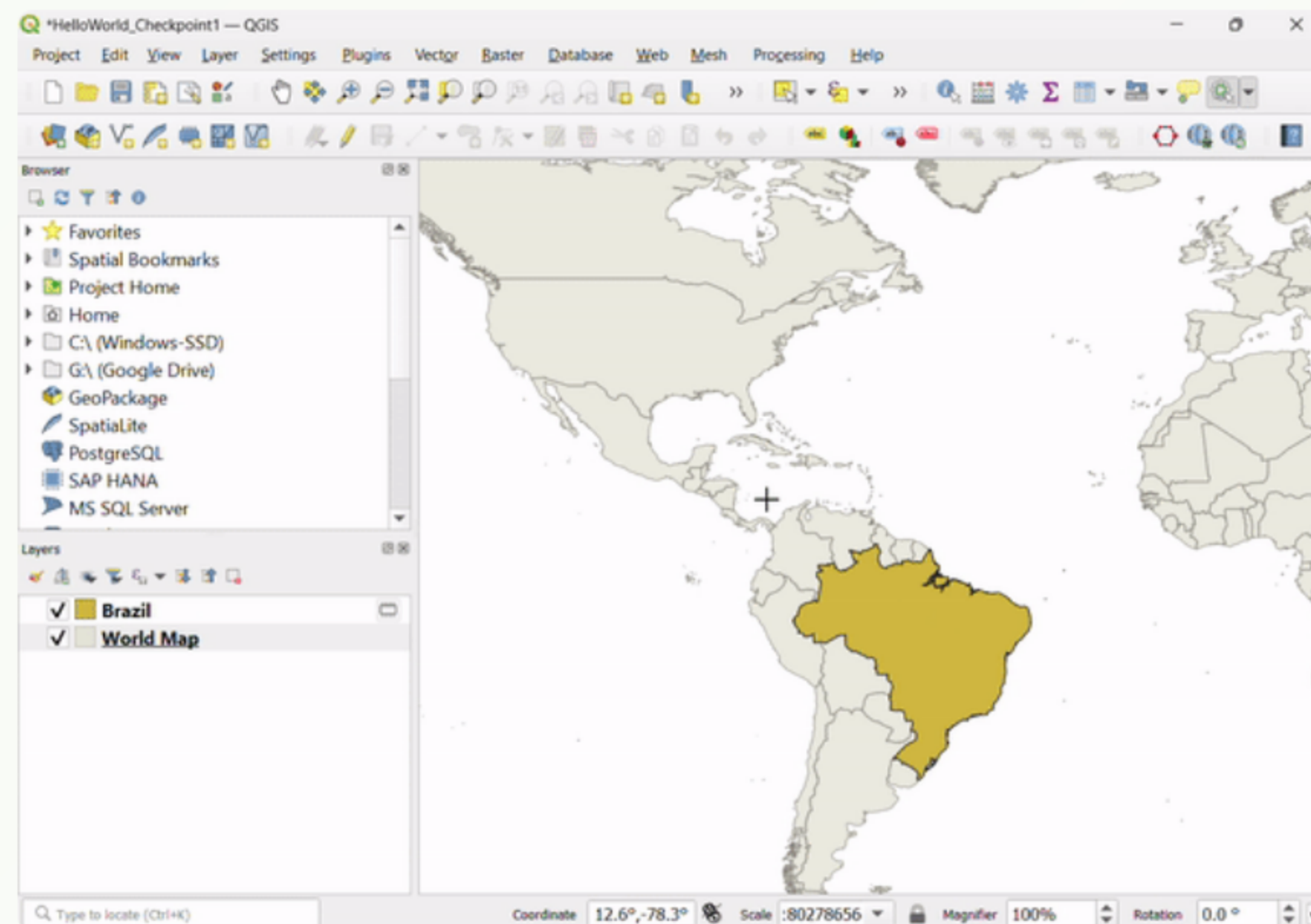
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

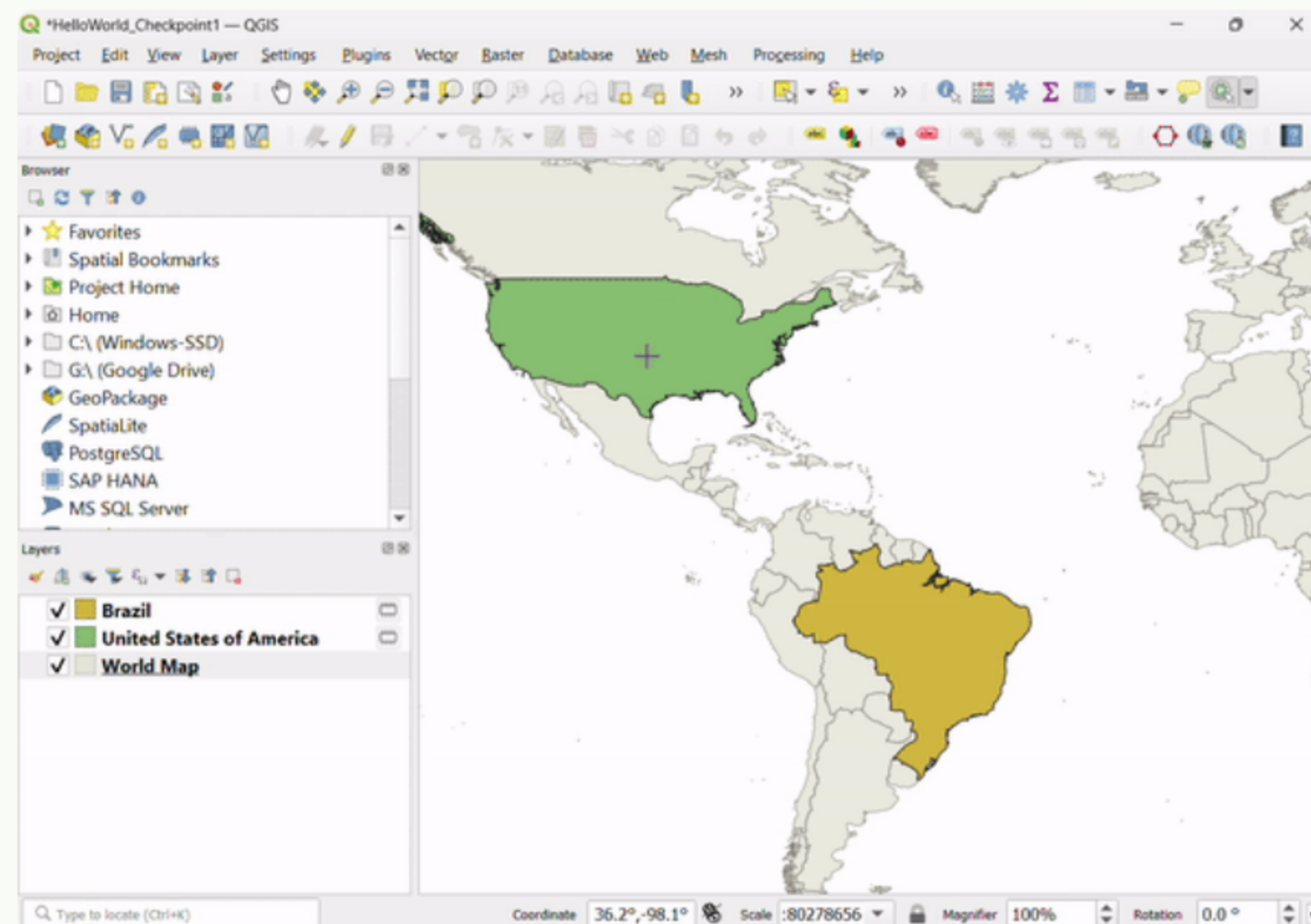
3 / 27

# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

# Przykłady akcji w QGIS

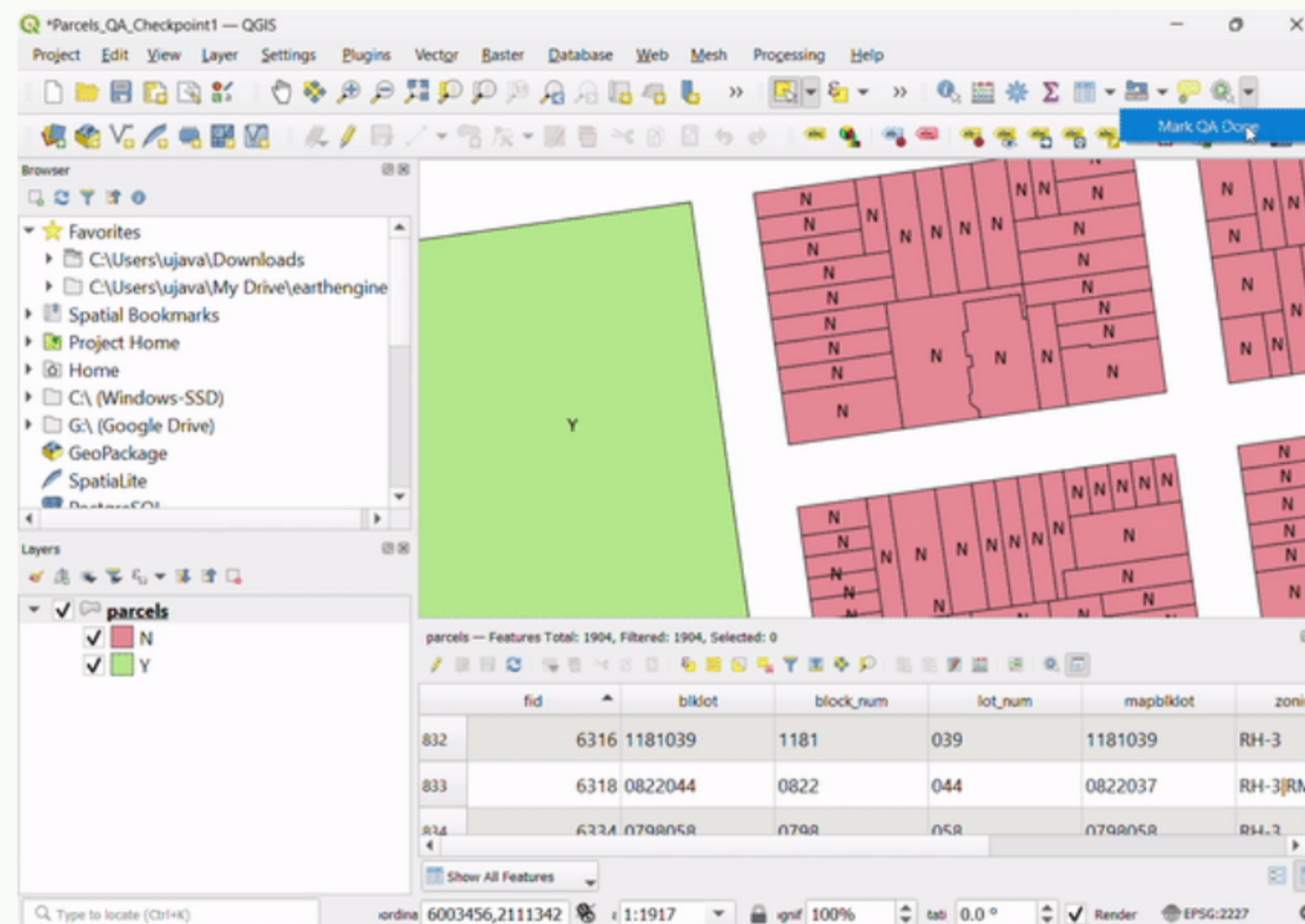


[źródło animacji: SpatialThoughts]

3 / 27

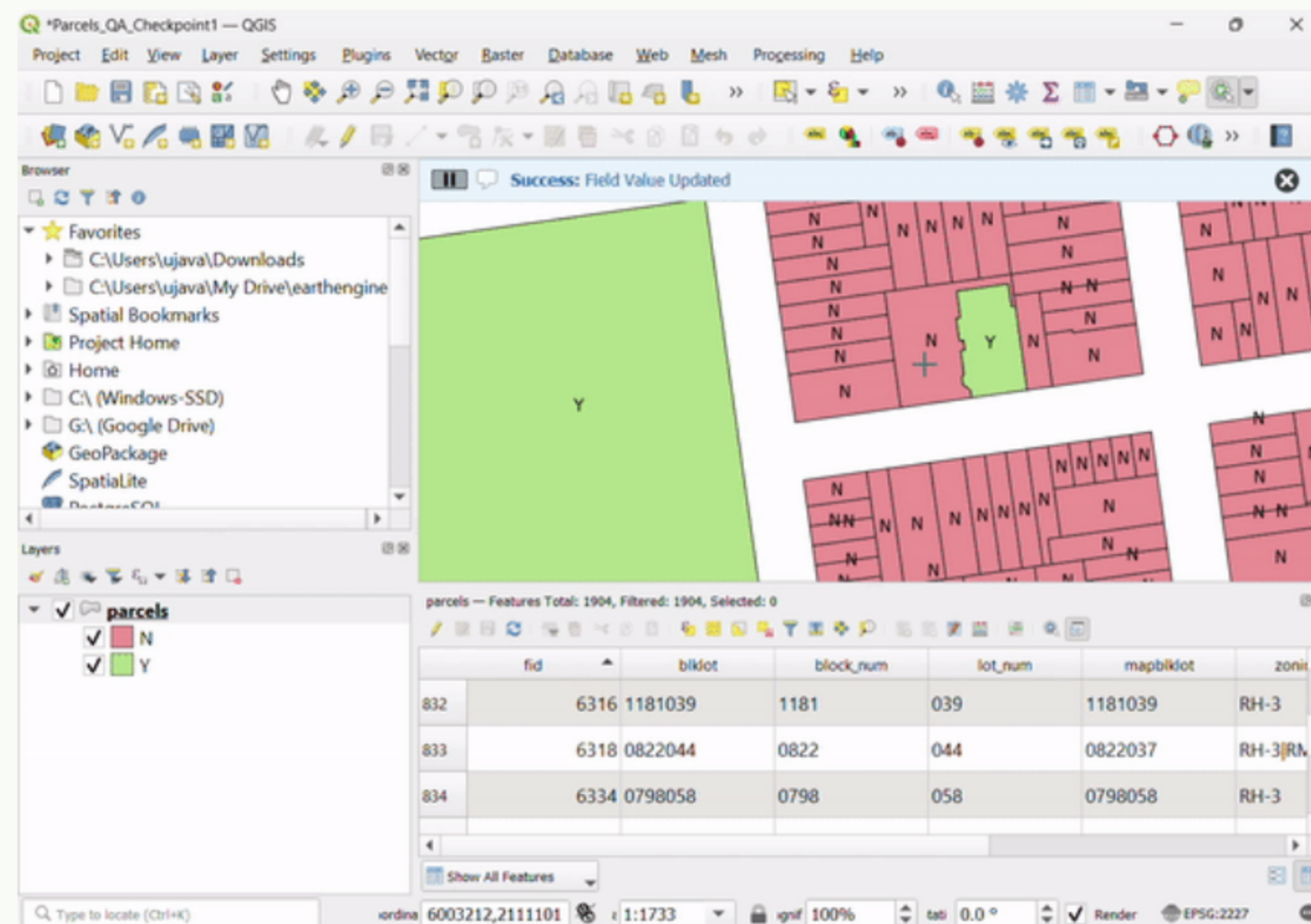


# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

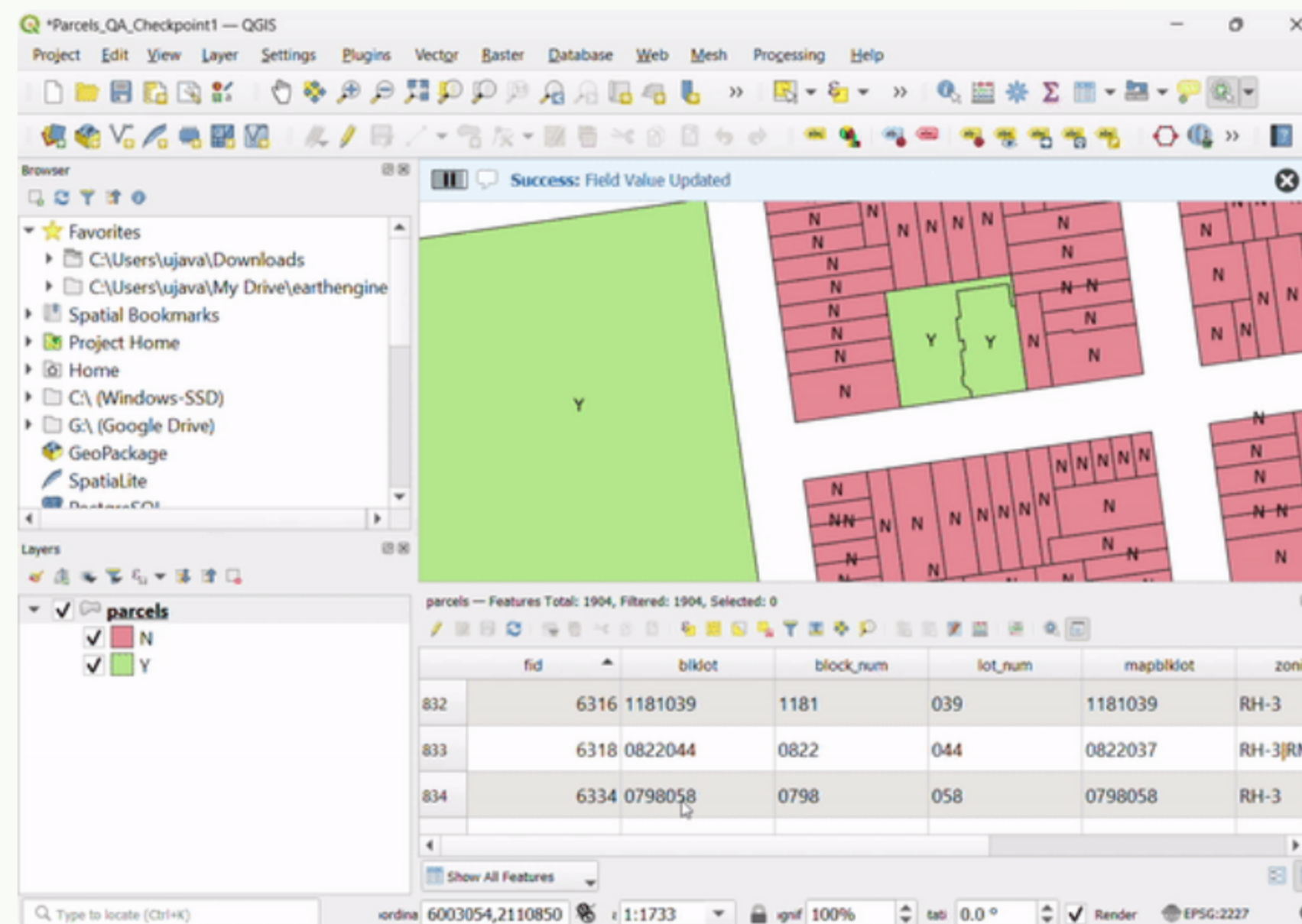
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

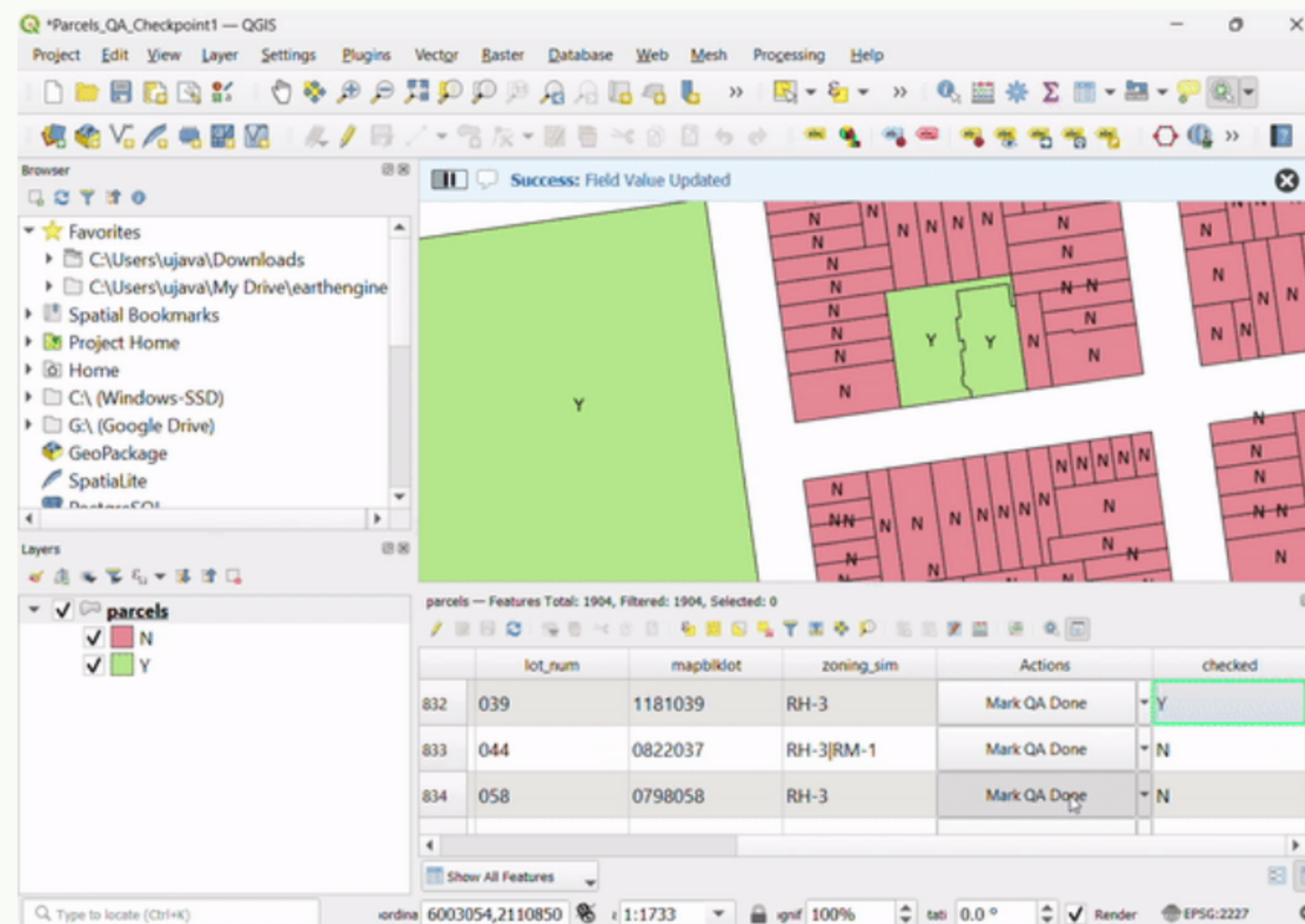


# Przykłady akcji w QGIS



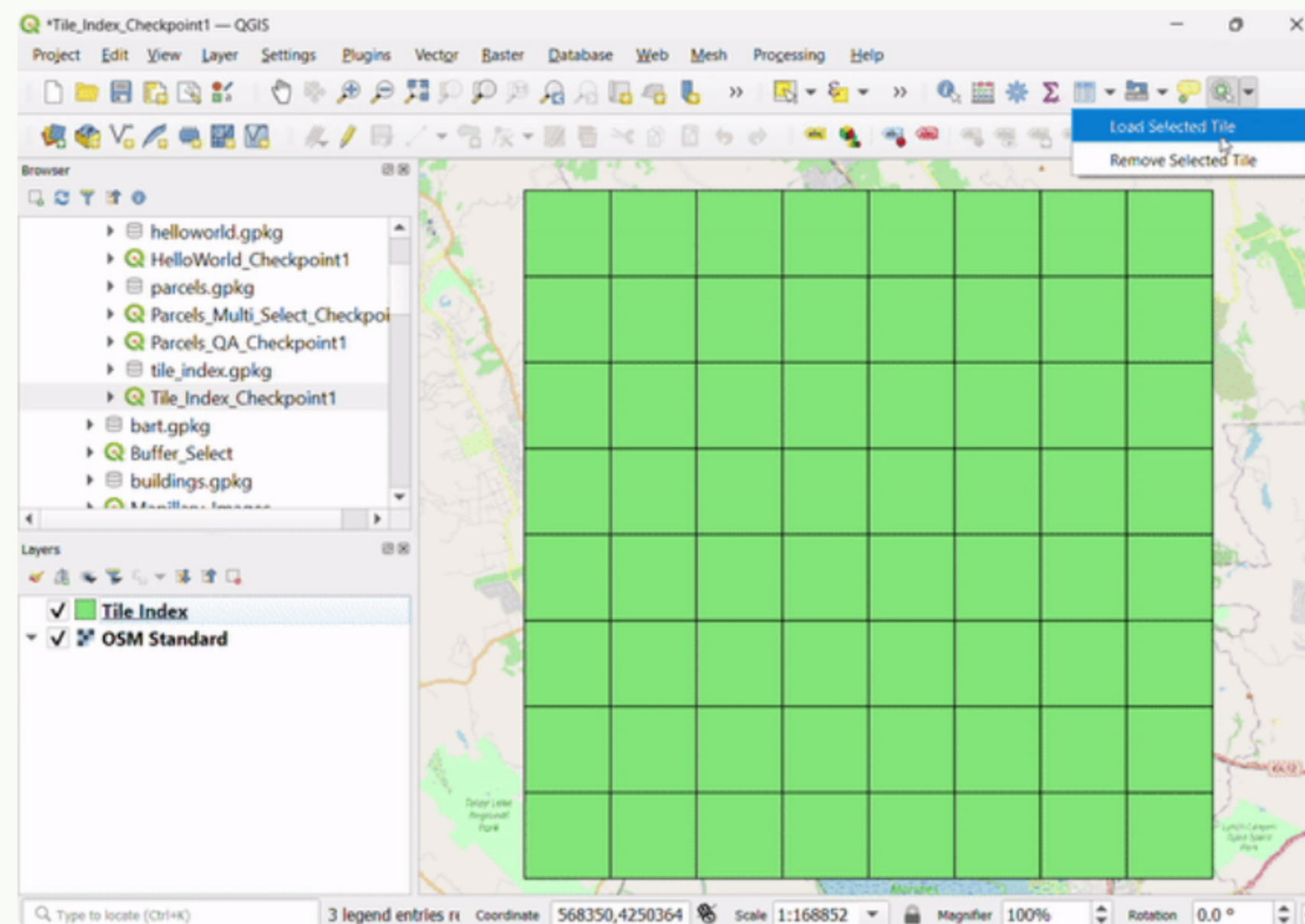
[źródło animacji: SpatialThoughts]

# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

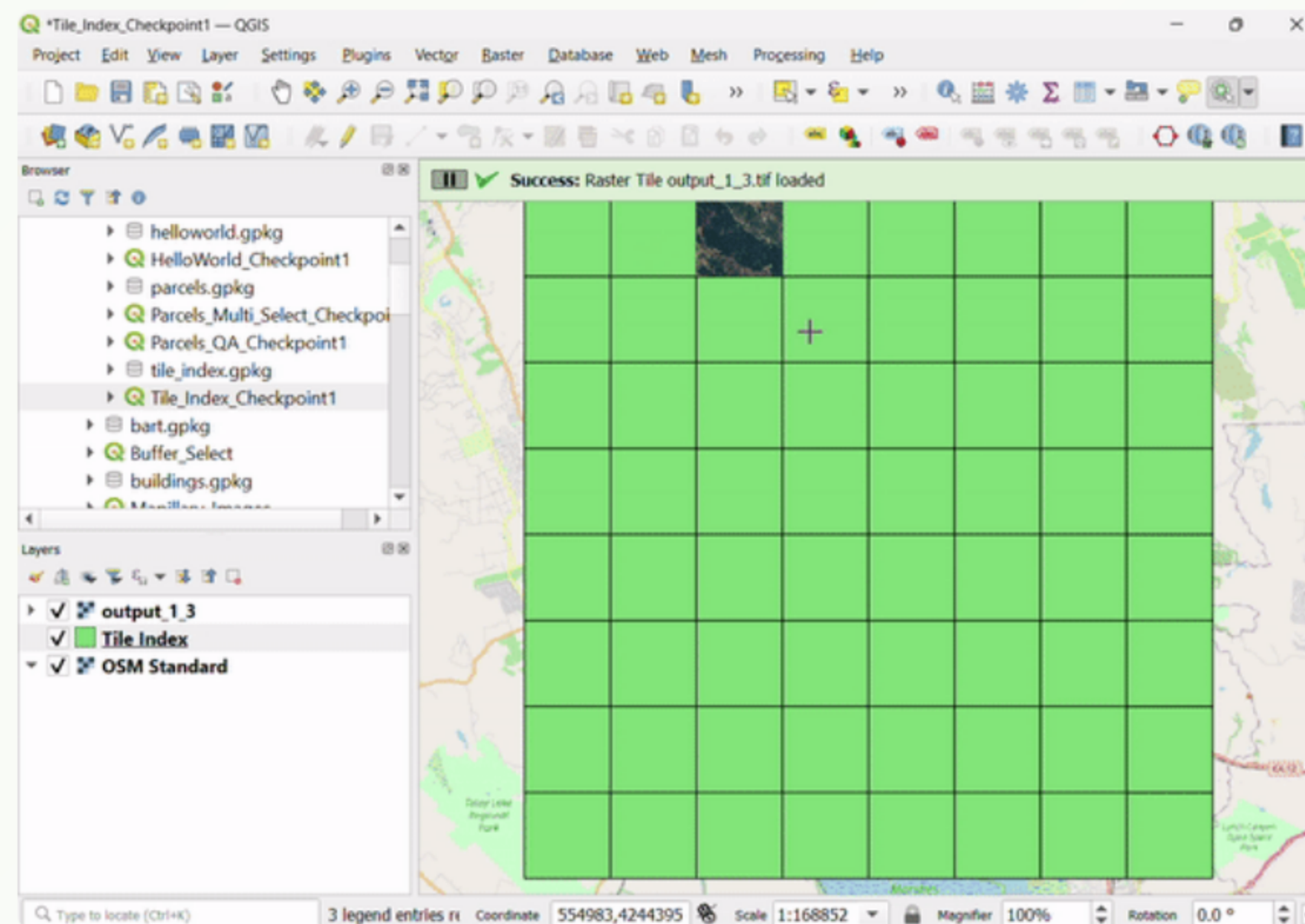
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



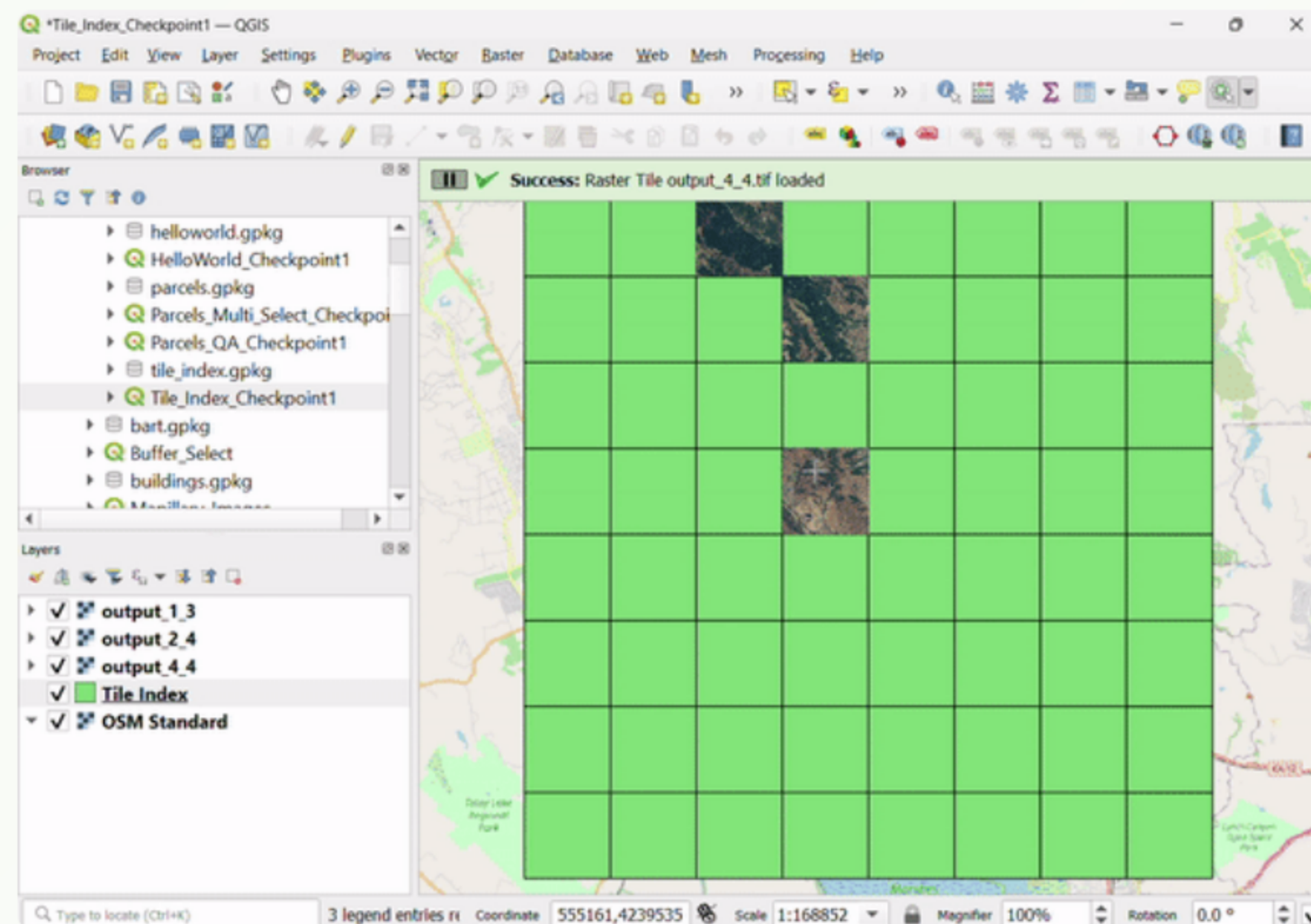
# Przykłady akcji w QGIS



[źródło animacji: SpatiaThoughts]

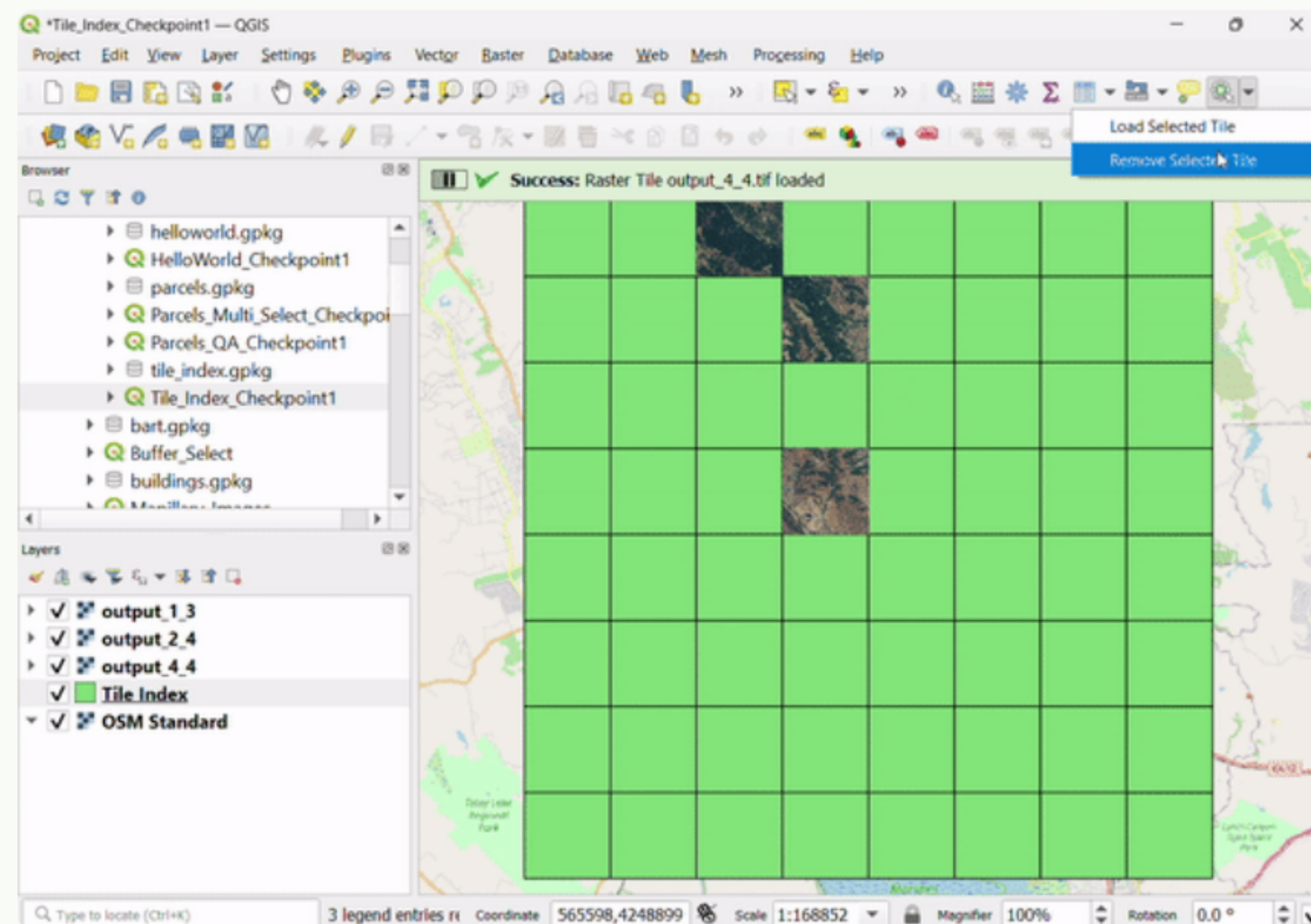


# Przykłady akcji w QGIS



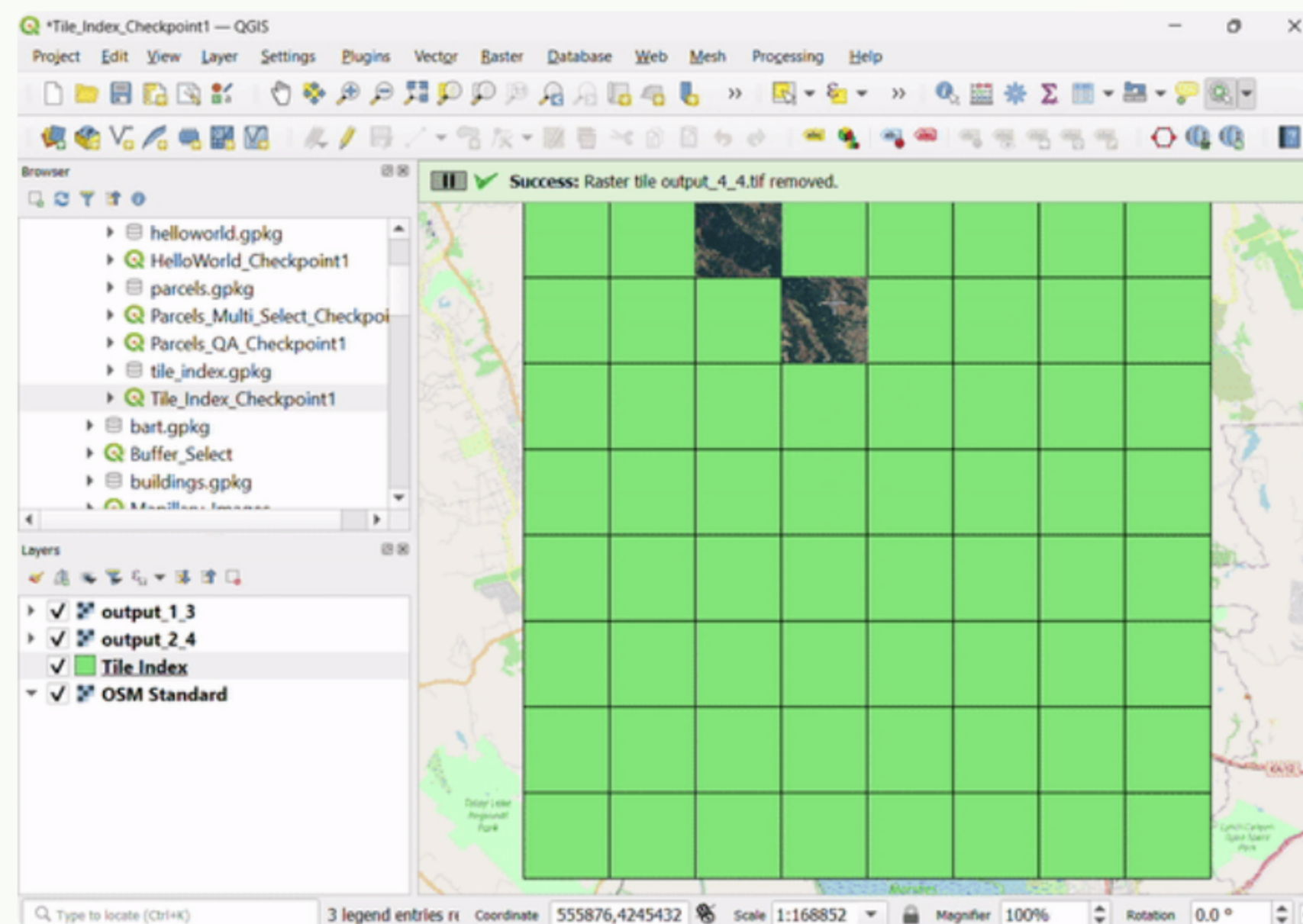
[źródło animacji: SpatialThoughts]

# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

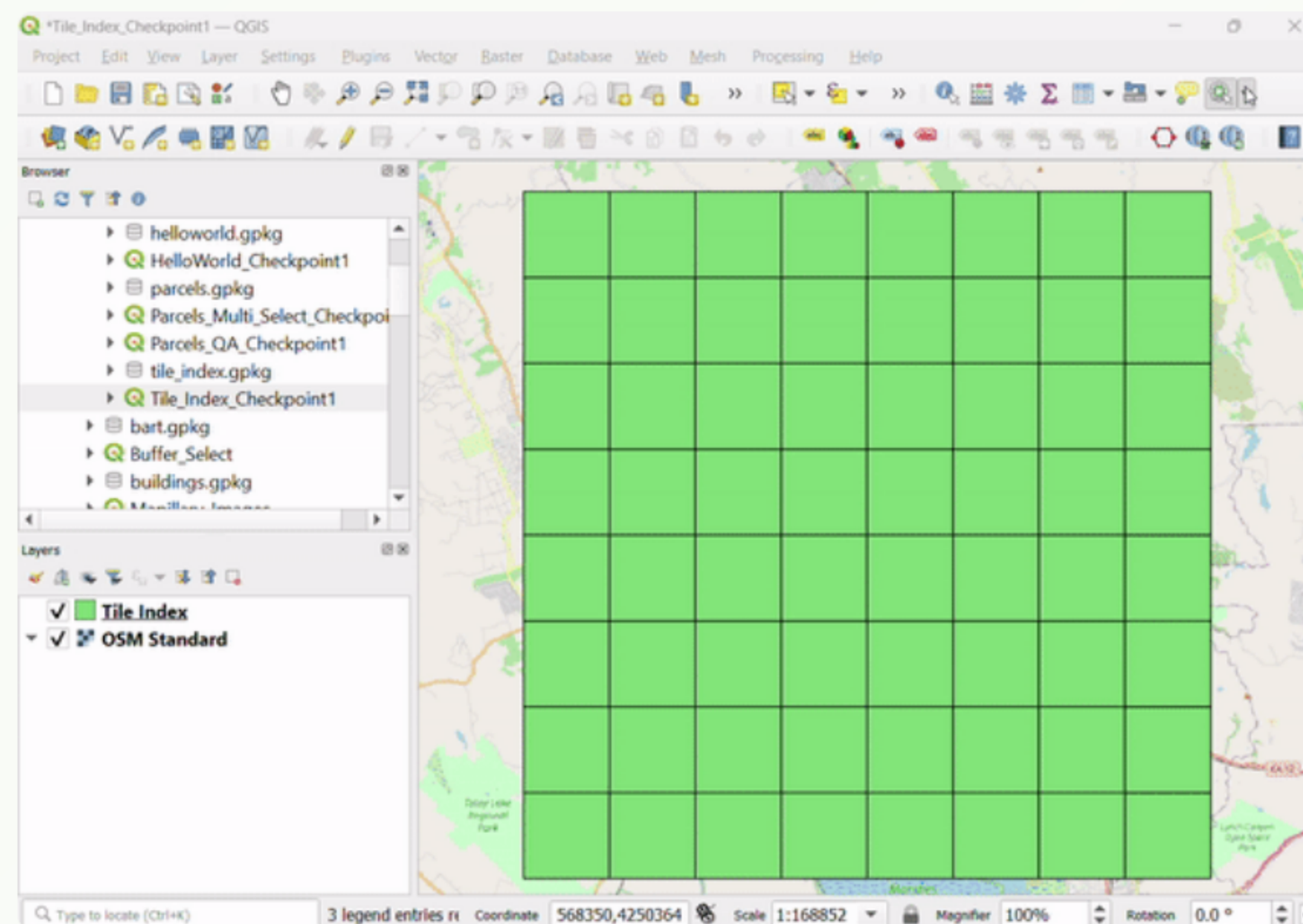
# Przykłady akcji w QGIS



[źródło animacji: SpatiaThoughts]

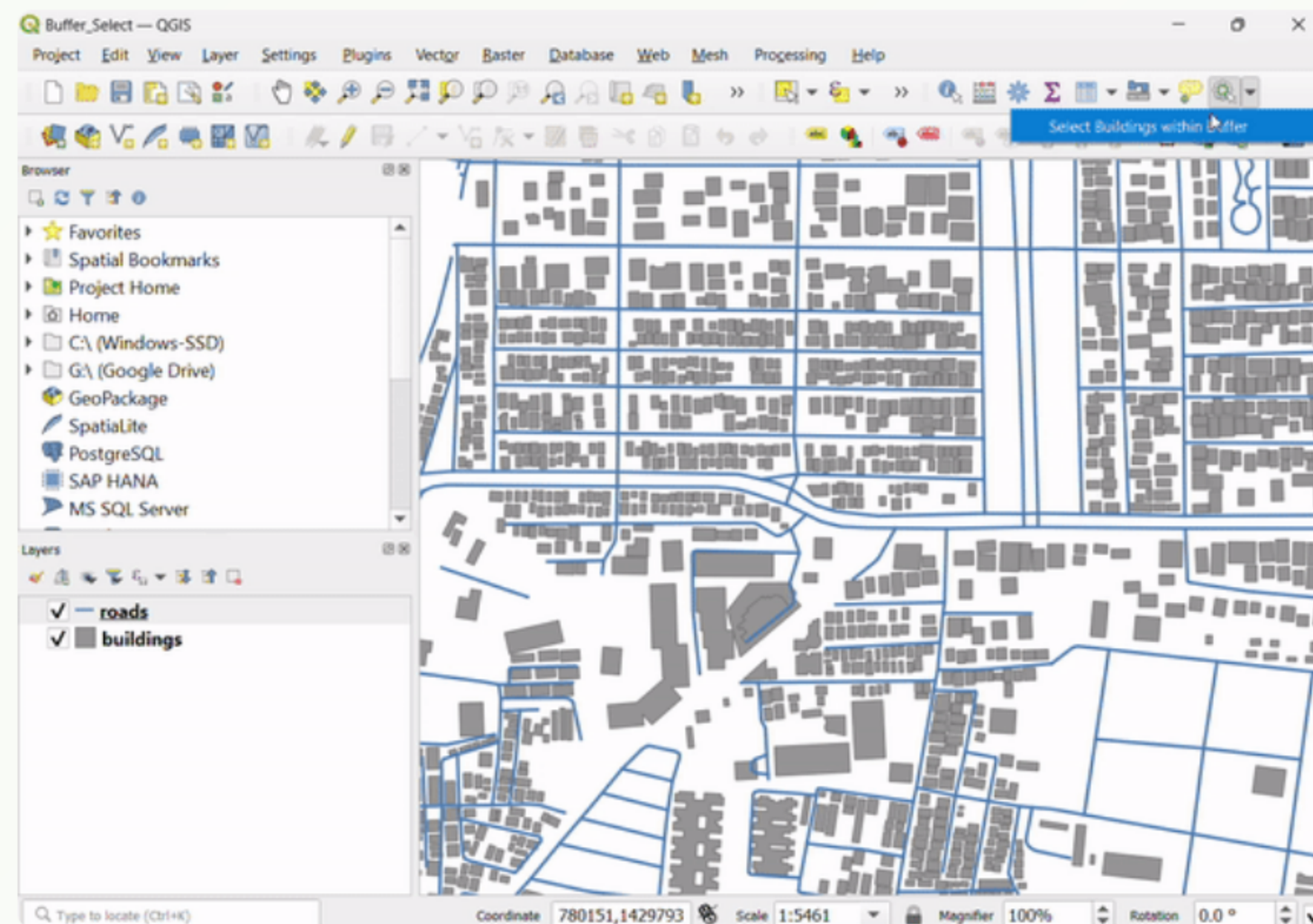


# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

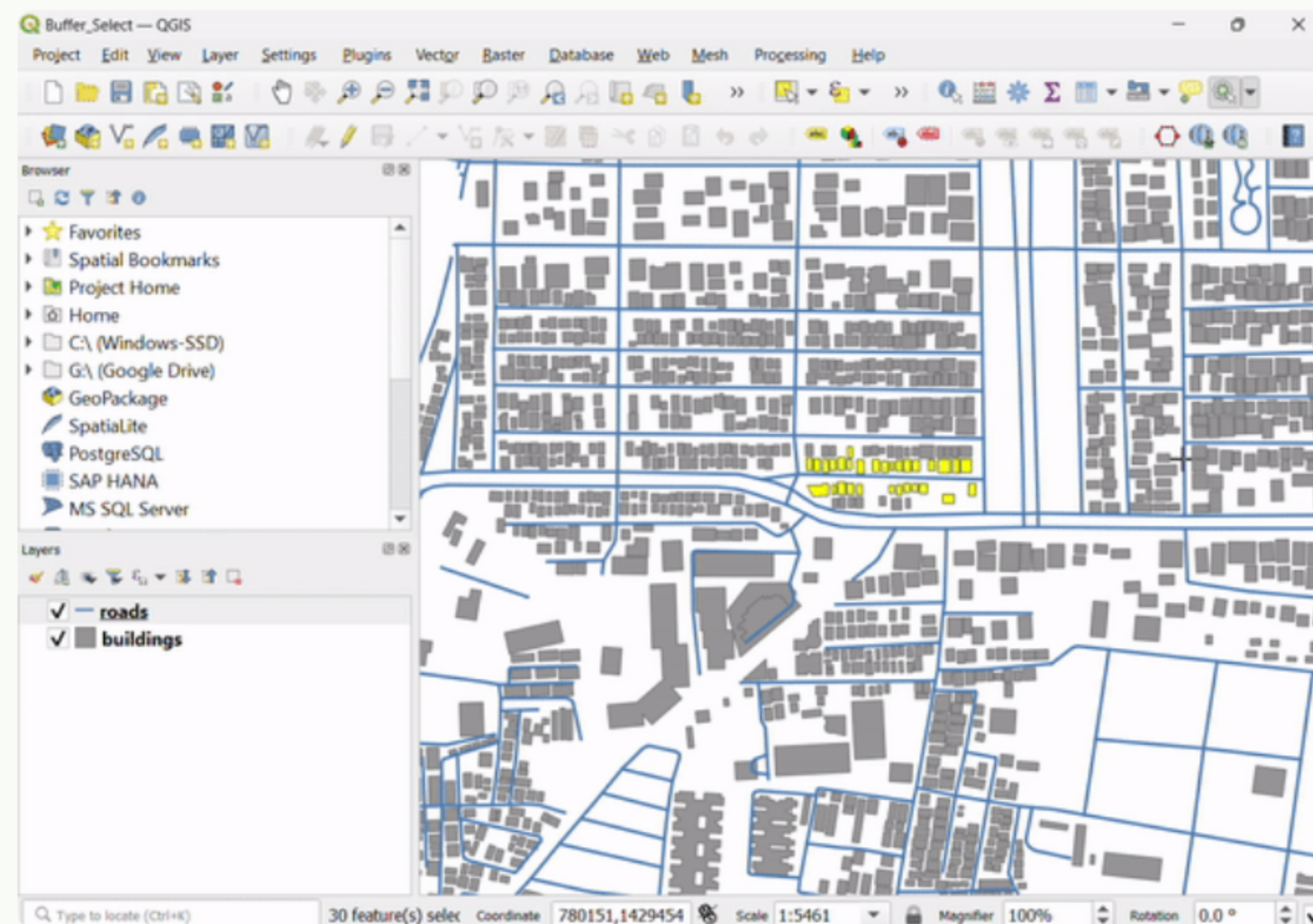
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



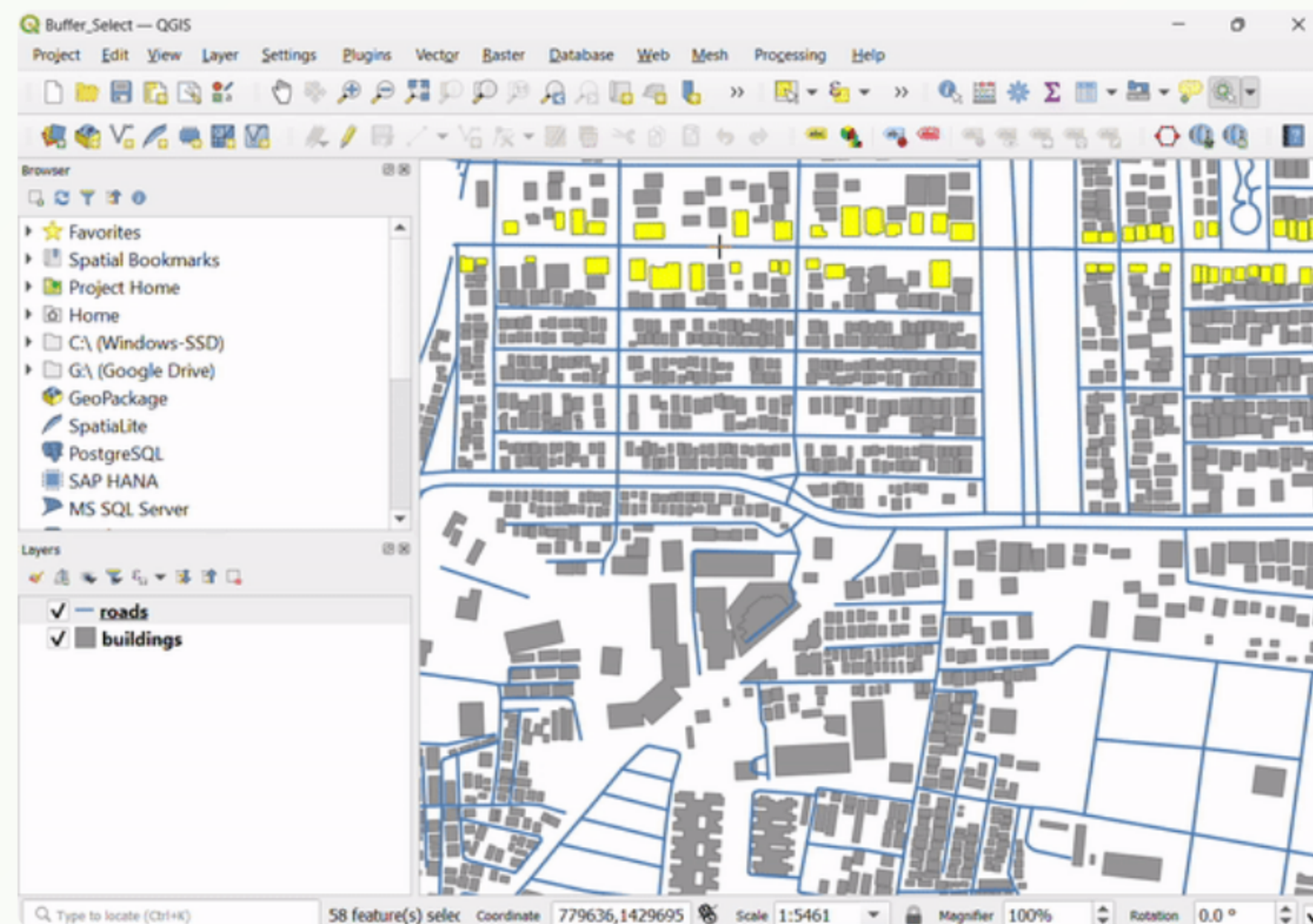
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

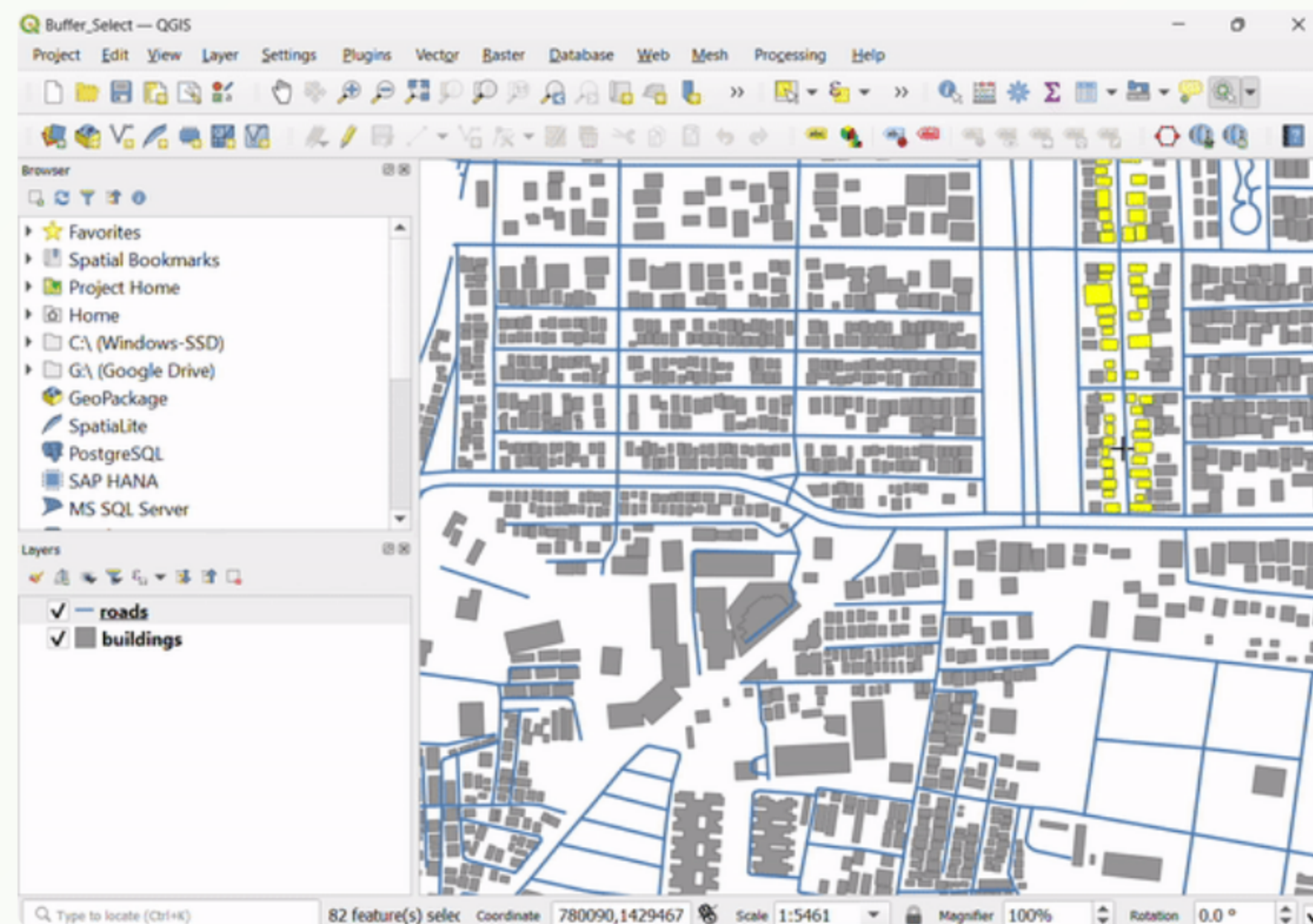


# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

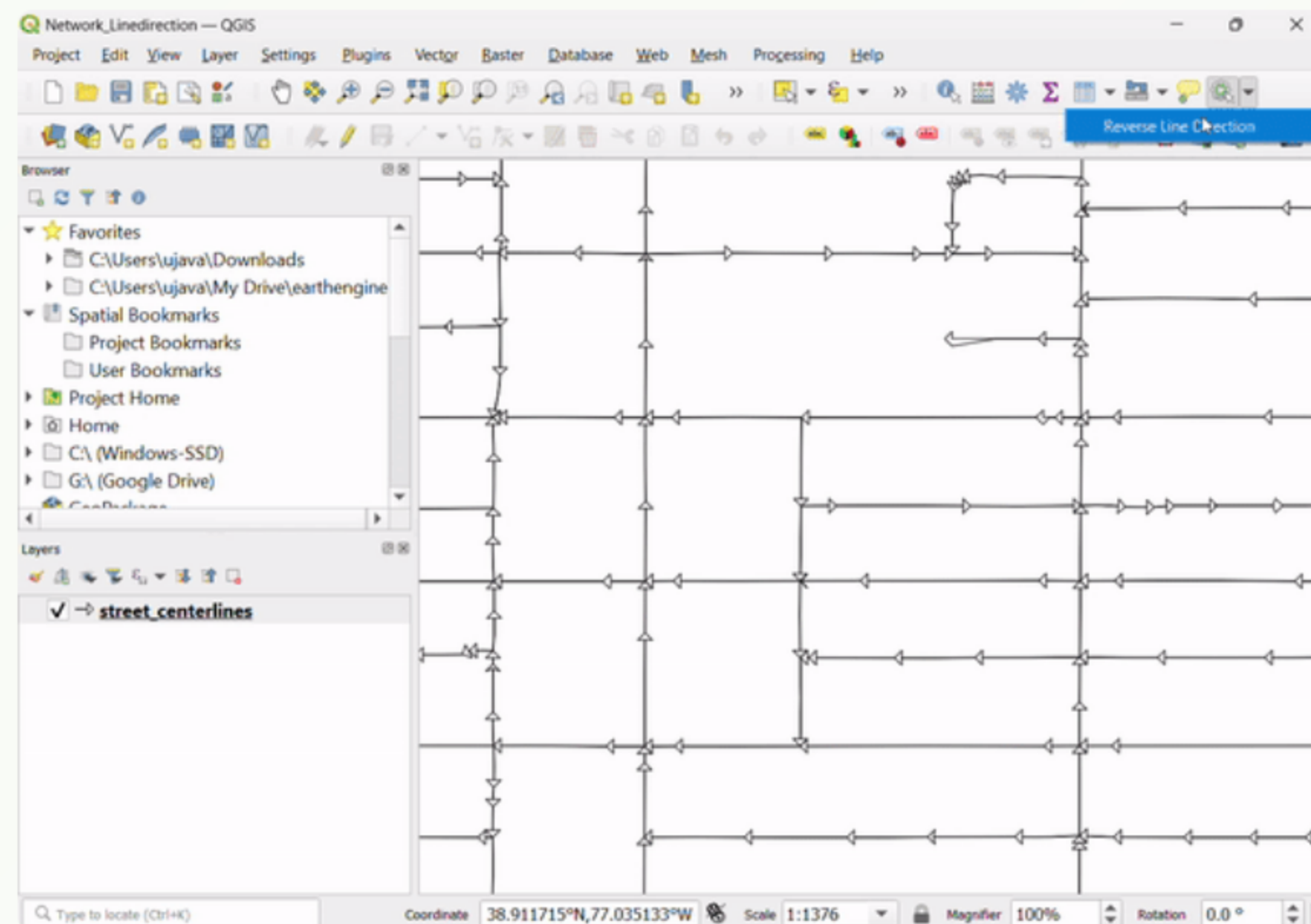
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



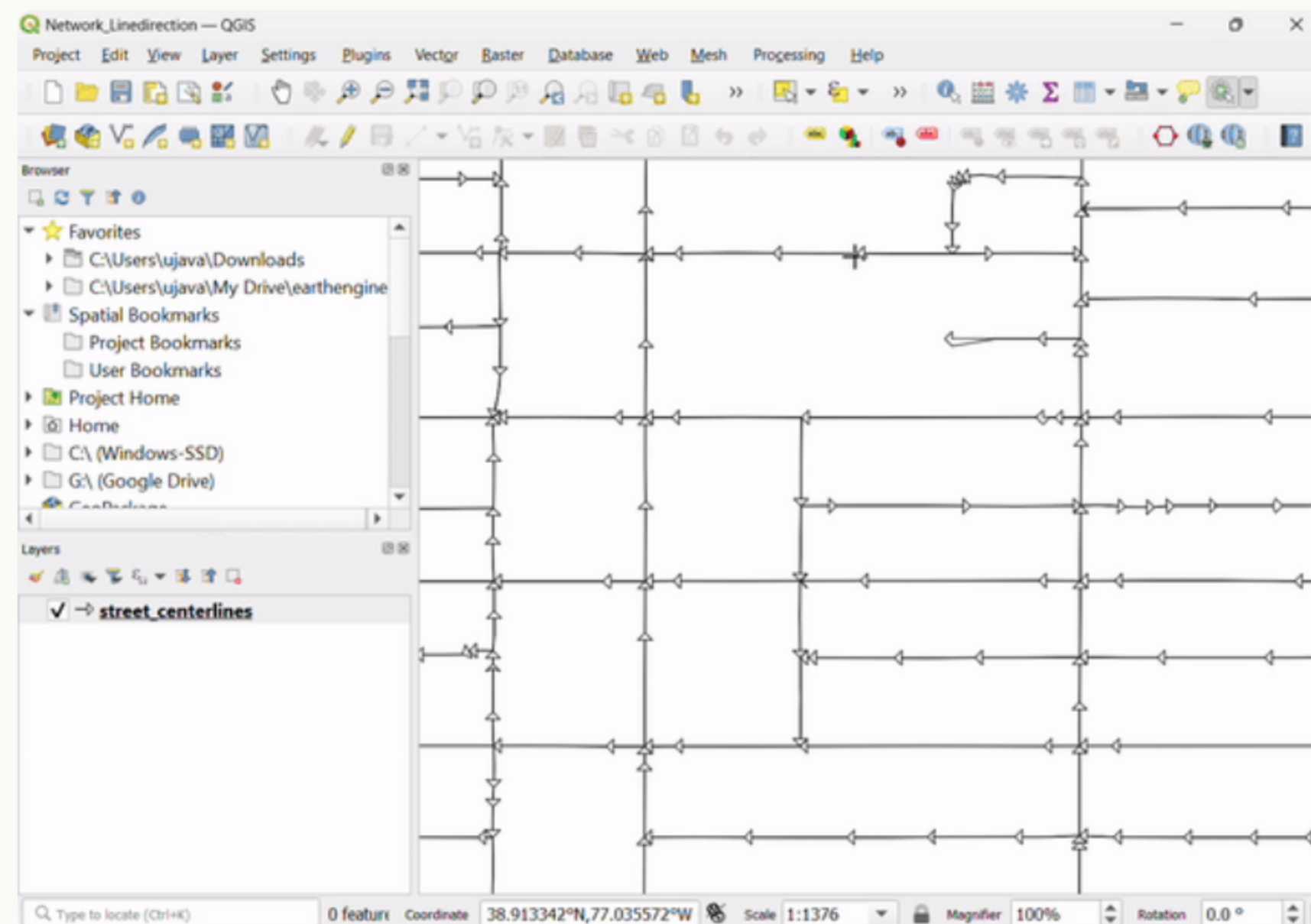
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

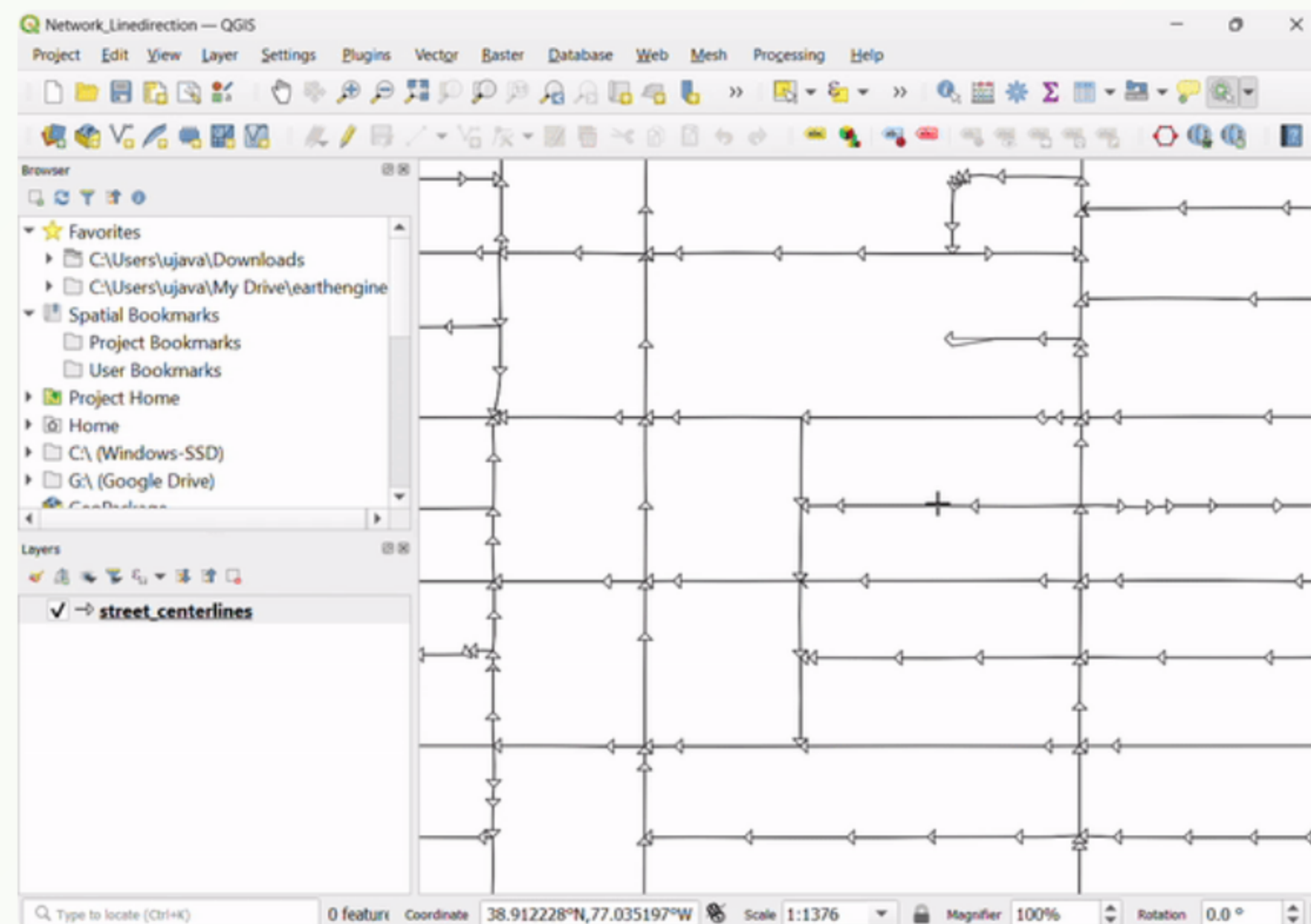


# Przykłady akcji w QGIS



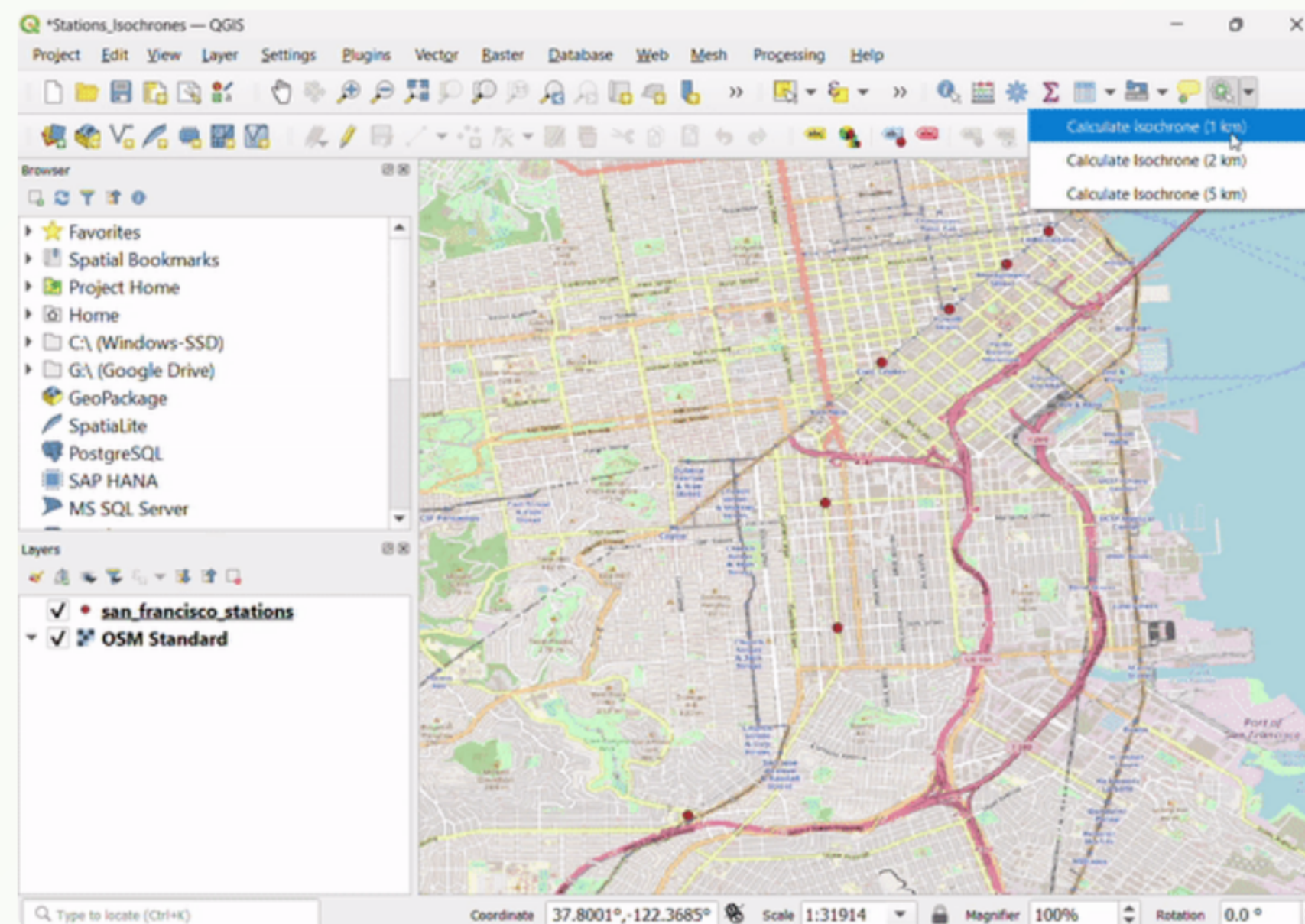
[źródło animacji: SpatialThoughts]

# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]

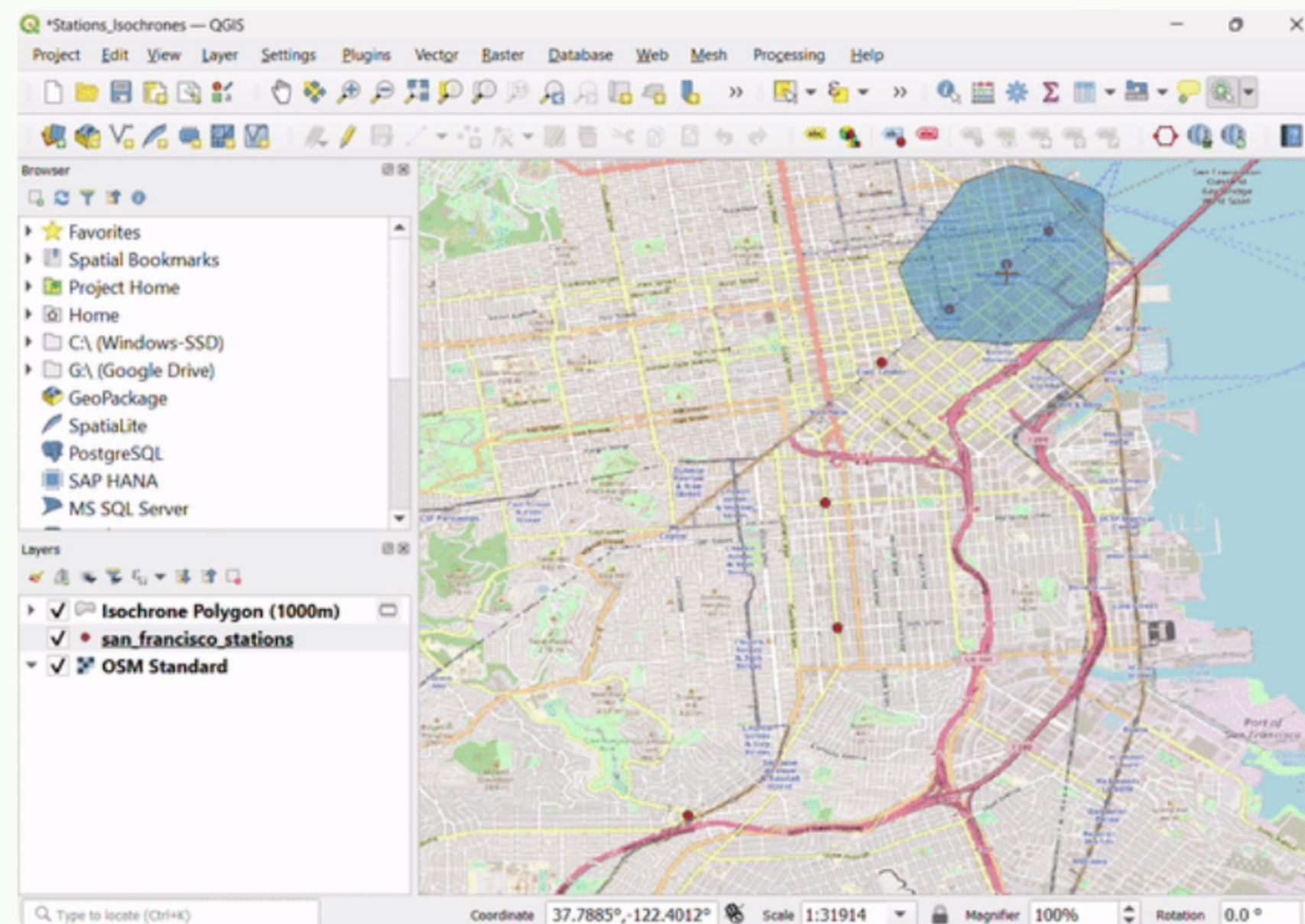
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



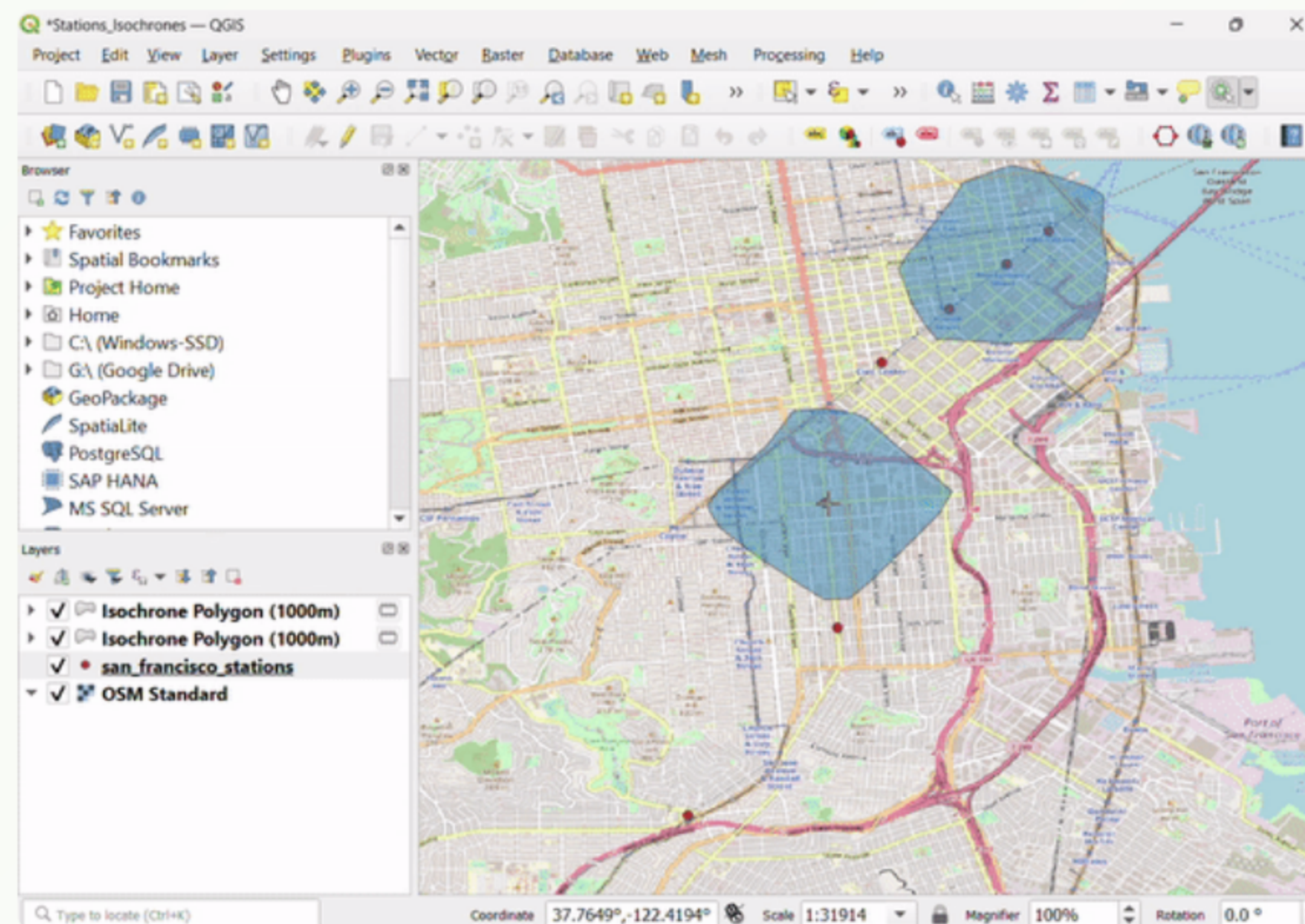
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



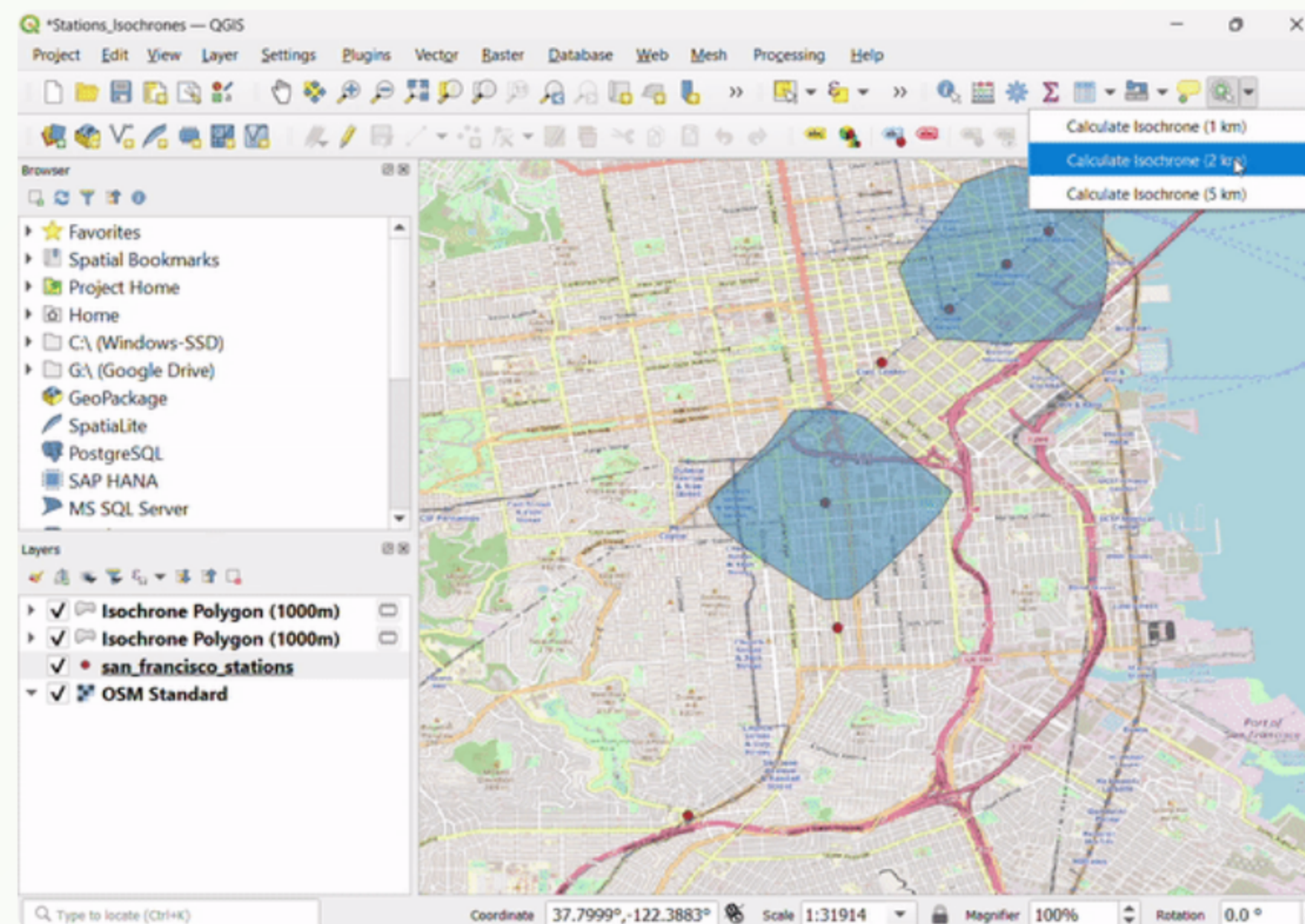
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



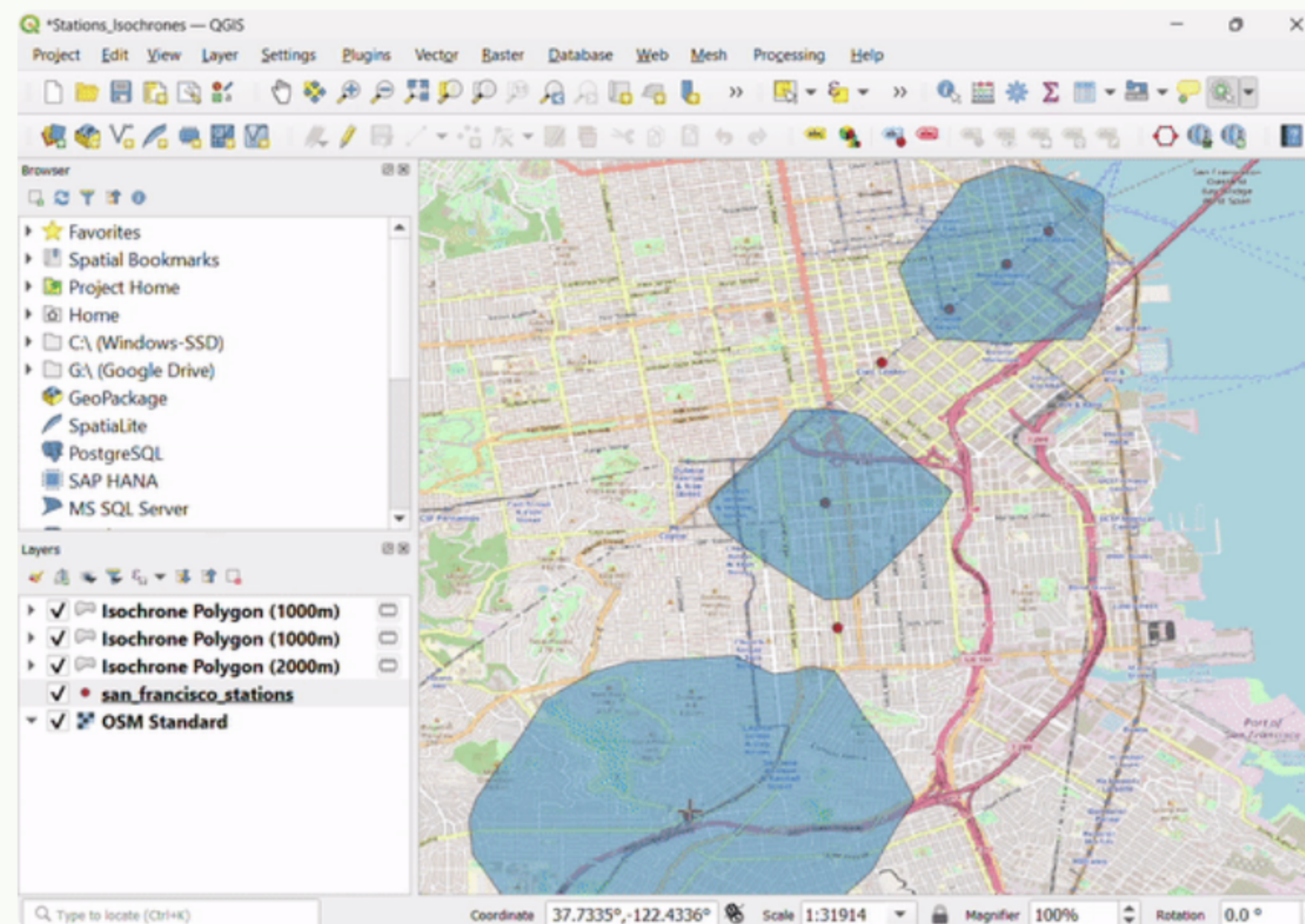
# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



# Przykłady akcji w QGIS



[źródło animacji: SpatialThoughts]



## Akcje to "mini-wtyczki"

- Wiele własnych funkcjonalności QGIS można zrealizować poprzez akcje, które działają jak mini-wtyczki.
- Akcje umożliwiają:
  - wykonywanie prostych operacji na obiektach
  - integrację z zewnętrznymi aplikacjami i usługami
  - wyświetlanie informacji o obiektach w formie okien dialogowych
  - wykonywanie dowolnego kodu Pythona w odniesieniu do warstwy, obiektu czy np. współrzędnych kliknięcia (nie musi być to interakcja z PyQGIS)
- Możliwe jest udostępnianie logiki akcji:
  - jako element składowy stylu warstwy
  - wraz z plikiem definicji warstwy

Dzięki temu można łatwo rozszerzać możliwości QGIS bez potrzeby tworzenia pełnoprawnych wtyczek.



# Typy akcji w QGIS

- **1 Generic** - uniwersalne akcje systemowe
- **2 Python** - skrypty w języku Python
- **3 Mac** - akcje dla systemu macOS
- **4 Windows** - akcje dla systemu Windows
- **5 Unix** - akcje dla systemów Unix/Linux
- **6 Open URL** - otwieranie adresów internetowych

Wybór typu akcji zależy od systemu operacyjnego i planowanej funkcjonalności.





# Konteksty akcji

- **1 Canvas (Widok mapy)**
  - Akcje uruchamiane z poziomu widoku mapy, np. przy kliknięciu na obiekt
  - Ma dostęp do warstwy (@layer) oraz atrybutów klikniętego obiektu
  - Ma dostęp do współrzędnych kliknięcia (@click\_x i @click\_y)
- **2 Feature (Obiekt)**
  - Pokazują się w tabeli atrybutów i formularzu obiektu
  - Ma dostęp do warstwy (@layer) oraz atrybutów klikniętego obiektu
- **3 Field (Pole)**
  - Pokazują się w menu PPM tabeli atrybutów i formularza obiektu
  - Ma dostęp do indeksu, nazwy i wartości pola (@field\_index, @field\_name, @field\_value)
- **4 Layer (Warstwa)**
  - Pokazują się w tabeli atrybutów. Działają na warstwie lub zaznaczeniu.
  - Ma dostęp do warstwy (@layer)
- **5 Form (Formularz)**
  - Pokazują się w Formularzu atrybutów przy projektowaniu metodą przeciągnij-i-upuść.
  - Ma dostęp do zmiennej @form



# Wykorzystanie parametrów w akcji

W akcjach można używać wartości atrybutów poprzez składnię:

```
[% "nazwa_pola" %]  
[% $id %] # ID obiektu  
[% @geometry %] # geometria obiektu  
[% $x %], [% $y %] # współrzędne
```

Można używać wyrażeń QGIS do dynamicznego generowania treści:

```
[% concat('Obiekt ID: ', $id, ' ma powierzchnię: ', round($area, 2), ' m²') %]
```

Akcje Python umożliwiają wykonywanie skryptów z dostępem do API QGIS

```
from qgis.utils import iface  
from qgis.PyQt.QtWidgets import QMessageBox  
  
field_value = '[% "nazwa_pola" %]'  
QMessageBox.information(iface.mainWindow(), 'Info', f'Wartość: {field_value}')
```

```
iface.messageBar().pushMessage(f'Wartość: {field_value}', level=Qgis.Info)
```



*- Ja proszę Pana jestem reżyserem kina AKCJI...*  
*- Noooo, to zobaczy Pan niezłą AKCJĘ!*

13 / 27





## Akcje w tabeli atrybutów

- Akcje te są dostępne jako przyciski w nagłówku tabeli atrybutów
- Można je uruchamiać dla wybranych obiektów
- Szczególnie przydatne przy pracy z wieloma obiektami
- Umożliwiają masowe przetwarzanie danych

### Przykład użycia:

Generowanie dokumentów dla wszystkich wybranych działek lub budynków

Dla przykładu stworzymy sobie teraz akcję generującą wnioski do urzędów gmin dla wybranych działek...

Akcja będzie działała na ogólnodostępnych danych: wartwie działek pobranej z EGiB oraz danych PRG i adresowych JST.

Pobierz dane [z tego linku](#).

<https://tomaszrychlicki.pl/static/files/akcje/akcje.7z>

# Akcja generująca wnioski do urzędów gmin



Stan plików w folderze przed uruchomieniem akcji

| /home/tomasz/Pulpit/akcje |      |         |  |
|---------------------------|------|---------|--|
| Nazwa                     | Roz. | Rozmiar |  |
| ^ [.]                     |      | <DIR>   |  |
| akcje_dane                | gpkg | 58.90 M |  |
| akcja_wniosek             | py   | 15.52 K |  |
| wniosek                   | txt  | 671     |  |

Szablon wniosku w pliku wniosek.txt

```
akcje > wniosek.txt
1      Łódź, {{data}} r.
2
3
4
5      {{nazwa_jednostki}}
6      {{adres_jednostki}}
7      {{kod_i_poczta}}
8
9
10     WNIOSEK
11
12
13     Zwracam się z wnioskiem o udostępnienie materiałów dotyczących {{działki}}. Dane te są niezbędne do wykonania niezwykle istotnej mapy w QGIS.
14     Proszę o pozytywne rozpatrzenie mojego wniosku.
15
16
17     Z poważaniem,
18     Tomasz Rychlicki
19
```

# Akcja generująca wnioski do urzędów gmin



## Kod akcji (z pliku akcja\_wniosek.py) - część 1

```
akcje > akcja_wniosek.py > ...
1  import textwrap
2  from collections import defaultdict
3  from datetime import date
4  from pathlib import Path
5  from qgis.core import QgsMessageLog
6  from qgis.PyQt.QtWidgets import QMessageBox
7  from qgis.utils import iface
8
9  FOLDER_Z_DANYMI = Path("/home/tomasz/Pulpit/akcje/")
10
11 layer = iface.activeLayer()
12 features = layer.getSelectedFeatures()
13
14 #? pobieram teryty z zaznaczonych obiektów do listy
15 teryty = [feat['teryt'] for feat in features if feat['teryt'] is not None]
16 # teryty = ["281411_2.0013.14/10", "281411_2.0012.119/19", "281411_2.0012.119/31", '281401_4.0002.100/2', '281401_4.0002.100/3'] #? przykładowe
    teryty do testów
17
18 #? generuję datę pisma w formacie dd.mm.rrrr
19 data_pisma = date.today().strftime("%d.%m.%Y")
20
21 #? słowniki z danymi o obrębach, gminach i powiatach oraz adresami JST - normalnie byłyby odczytane z bazy danych lub pliku konfiguracyjnego
22 słownik_obrebow = {
23     "281401_4.0001": "Miasto Barczewo 1",
24     "281401_4.0002": "Miasto Barczewo 2",
25     "281401_4.0003": "Miasto Barczewo 3",
26     "281401_5.0001": "Barczewko",
27
28     "281412_2.0014": "Wysokie",
29 }
30
31 słownik_gmin = {
32     "281401": "Barczewo",
33     "281412": "Świątki",
34 }
35
36 słownik_powiatow = {"2814": "olsztyński"}
37
38 adresy = {
39     "281400": ("Starostwo Powiatowe w Olsztynie", "Plac Bema 5", "10-516", "Olsztyn"),
40     "281401": ("Urząd Miejski Barczewo", "pl. Ratuszowy 1", "11-010", "Barczewo"),
41     "281402": ("Urząd Miejski w Biskupcu", "al. Niepodległości 2", "11-300", "Biskupiec"),
42
43     "281412": ("Urząd Gminy Świątki", "Świątki 87", "11-008", "Świątki"),
44 }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```



# Akcja generująca wnioski do urzędów gmin



## Kod akcji (z pliku akcja\_wnioszek.py) - część 2

```
akcje > akcja_wnioszek.py > ...
356
357     #? funkcja do podziału linii dłuższych niż 80 znaków, z zachowaniem tabulacji i bez dzielenia słów
358     def wrap_line(line, width=80):
359         if line.rstrip() != line: # zachowaj wiodące tabulacje/spacje
360             indent = line[:len(line) - len(line.rstrip())]
361         else:
362             indent = ''
363         return '\n'.join(textwrap.wrap(line, width=width, break_long_words=False, break_on_hyphens=False, initial_indent=indent,
364                                     subsequent_indent=indent)) or line
365
366     #! grupowanie działek gminami i obrębami
367     dzialki_gminami = defaultdict(lambda: defaultdict(list))
368     for teryt in teryty:
369         kod_gminy = teryt[:6] # pierwsze 8 znaków - gmina
370         kod_obrebu = teryt[:13] # pierwsze 13 znaków - obręb
371         dzialki_gminami[kod_gminy][kod_obrebu].append(teryt)
372
373     #! iteruję po gminach, bo dla każdej gminy tworzę nowy wniosek
374     for gmina in dzialki_gminami:
375         print('-'*70) #? linia do oddzielenia printów dla poszczególnych gmin w konsoli Pythona
376         #! lista słów dla stringu opisującego działki
377         opis_dzialek = ''
378         #? w zależności od liczby działek w gminie, dodajemy odpowiednie słowo
379         if len(dzialki_gminami[gmina]) > 1 or any(len(dzialki) > 1 for dzialki in dzialki_gminami[gmina].values()):
380             opis_dzialek += "działek"
381         else:
382             opis_dzialek += "działki"
383         #? dodajemy obręby i ich nazwy oraz numery działek w obrębie
384         for obreb, dzialki in dzialki_gminami[gmina].items():
385             opis_dzialek += f"{' ', '.join([d.split('.')[1] for d in dzialki])}"
386             opis_dzialek += f" w obrębie {sloownik_obrebow[obreb]}"
387             opis_dzialek += " i"
388         opis_dzialek = opis_dzialek[:-2] #? obcinam ostatnie 2 znaki (ostatnie " i")
389         opis_dzialek += f" gmina {sloownik_gmin[gmina]}"
390         opis_dzialek += f" powiat {sloownik_powiatow[gmina[:4]]}"
```

# Akcja generująca wnioski do urzędów gmin



## Kod akcji (z pliku akcja\_wnioski.py) - część 3

```
akcje > akcja_wnioski.py > ...
390
391 with open(FOLDER_Z_DANYMI / "wniosek.txt", "r") as szablon:
392     tekst_wniosku = szablon.read()
393     tekst_wniosku = tekst_wniosku.replace("{{data}}", data_pisma)
394     tekst_wniosku = tekst_wniosku.replace("{{nazwa_jednostki}}", adresy[gmina][0])
395     tekst_wniosku = tekst_wniosku.replace("{{adres_jednostki}}", adresy[gmina][1])
396     tekst_wniosku = tekst_wniosku.replace("{{kod_i_poczta}}", f"{adresy[gmina][2]} {adresy[gmina][3]}")
397     tekst_wniosku = tekst_wniosku.replace("{{działki}}", opis_działek)
398
399     #? skracam zawartość do 80 znaków, aby nie przekraczać szerokości linii
400     gotowy_tekst_wniosku = '\n'.join(sum([wrap_line(l).split('\n') for l in tekst_wniosku.splitlines()], []))
401     with open(FOLDER_Z_DANYMI / f"wniosek_{slovník_gmin[gmina]}_{data_pisma}.txt", "w") as plik_wynikowy:
402         plik_wynikowy.write(gotowy_tekst_wniosku)
403
404     print(gotowy_tekst_wniosku)
405     #? dodanie info do MessageBar
406     iface.messageBar().pushMessage("Wnioski", f"Pomyślnie wygenerowano wniosek dla gminy: {slovník_gmin[gmina]}", level=Qgis.Success, duration=5)
407     #? dodanie info do logu QGIS
408     QgsMessageLog.logMessage(f"Pomyślnie wygenerowano wniosek dla gminy: {slovník_gmin[gmina]}", 'AkcjaGenerujWnioski', level=Qgis.Success)
409
410     QMessageBox.information(iface.mainWindow(), 'Info', f"Pomyślnie wygenerowano wnioski dla gmin: {' '.join([slovník_gmin[gmina] for gmina in
411     działki_gminami])}")
```



*Spokojnie, zaraz się rozkręci...*

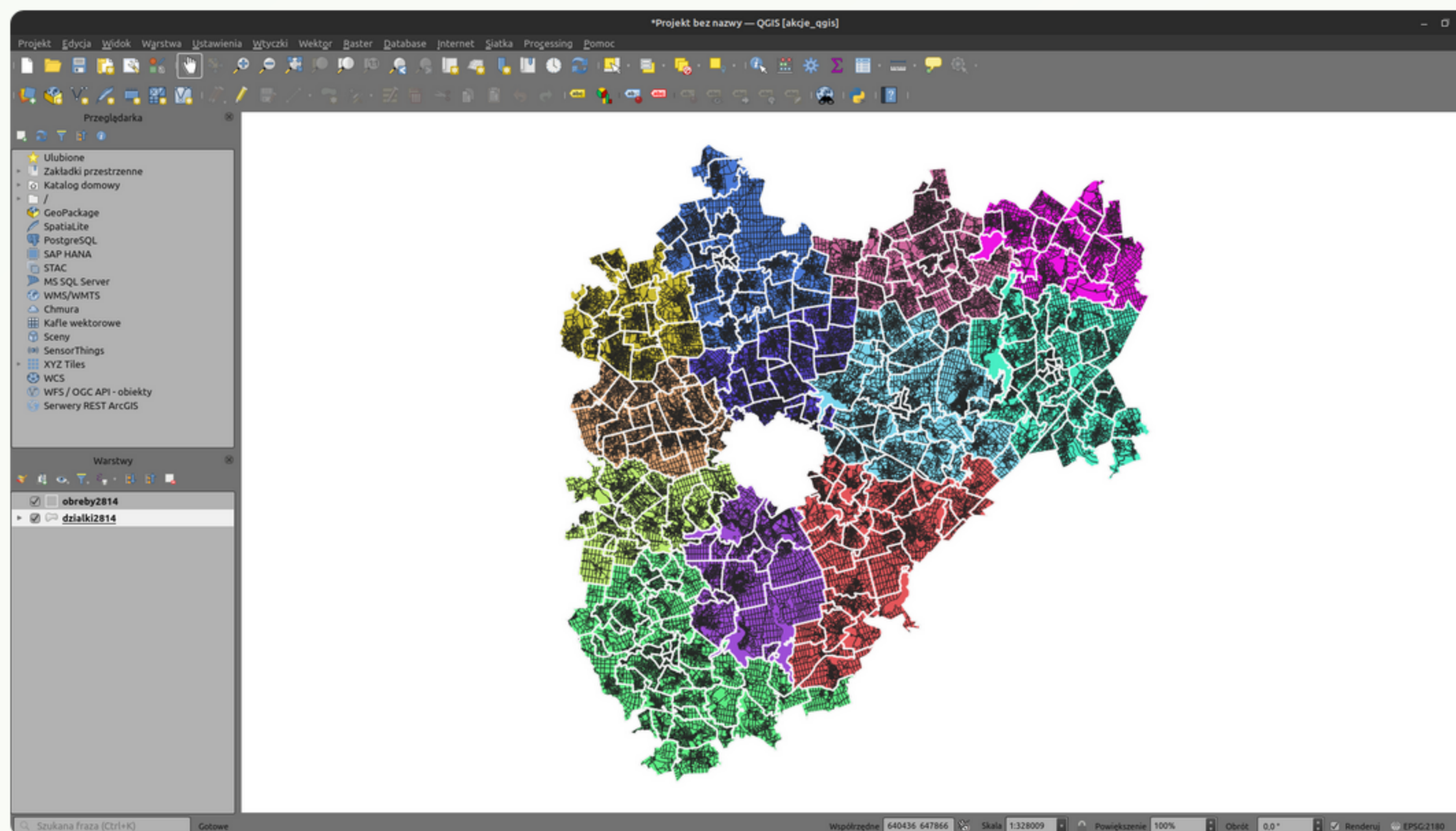
19 / 27



# Akcja generująca wnioski do urzędów gmin



Okno QGIS po wczytaniu danych z pliku akcje\_dane.gpkg

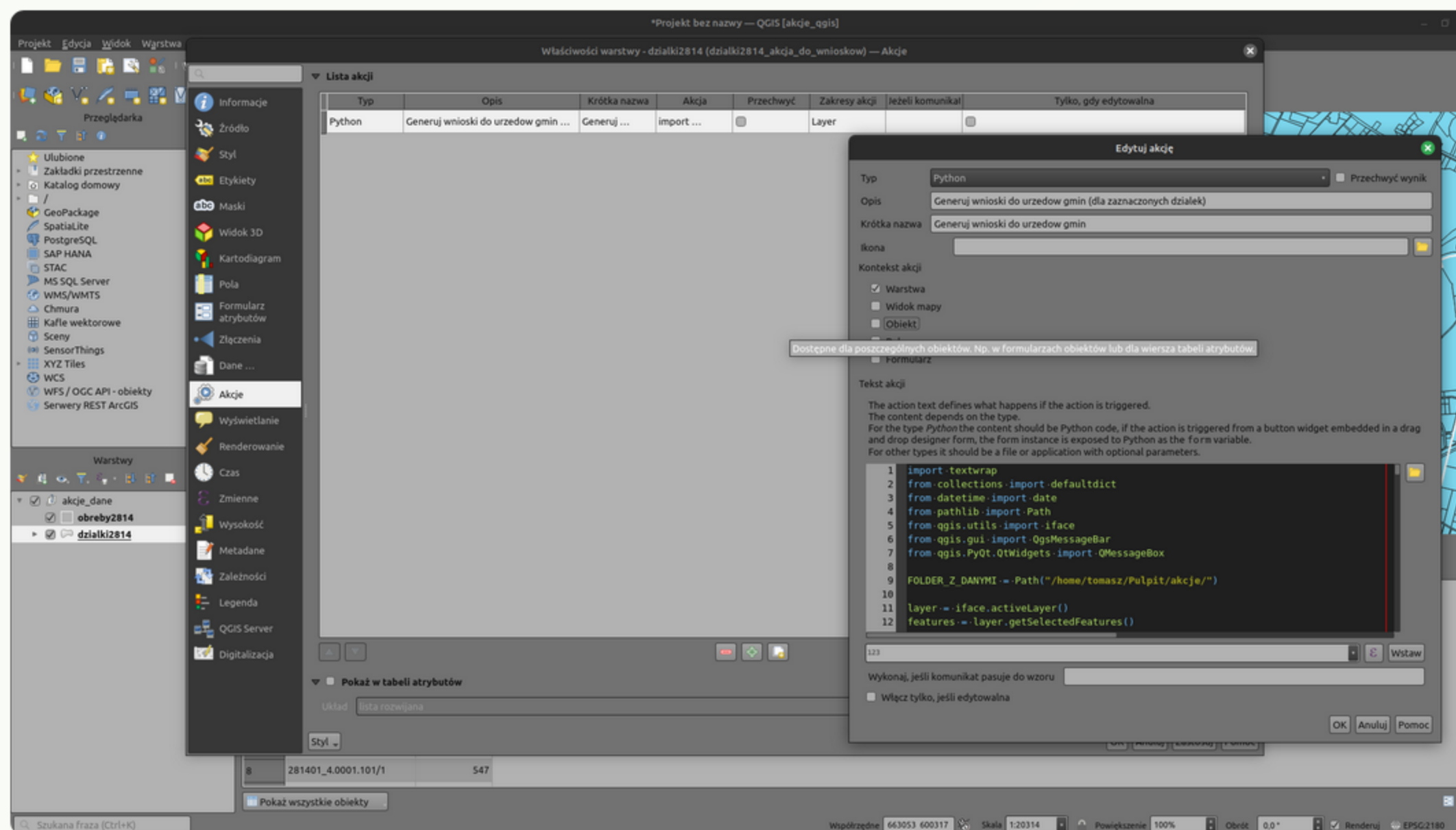


20 / 27

# Akcja generująca wnioski do urzędów gmin



## Edycja akcji



21 / 27

# Akcja generująca wnioski do urzędów gmin



## Wykorzystanie akcji dla zaznaczonych działek

Przeglądarka

- Ulubione
- Zakładki przestrzenne
- Katalog domowy
- /
- GeoPackage
- Spatialite
- PostgreSQL
- SAP HANA
- STAC
- MS SQL Server
- WMS/WMTS
- Chmura
- Kafle wektorowe
- Sceny
- SensorThings
- XYZ Tiles
- WCS
- WFS / OGC API - obiekty
- Serwery REST ArcGIS

Warstwy

- obreby2814
- działki2814

działki2814 — Łącznie obiektów: 171519, odfiltrowanych: 171519, wybranych: 8

|   | teryt               | fid  |
|---|---------------------|------|
| 1 | 281401_4.0001.1     | 2269 |
| 2 | 281401_4.0001.10    | 2263 |
| 3 | 281401_4.0001.100/4 | 552  |
| 4 | 281401_4.0001.100/5 | 2300 |
| 5 | 281401_4.0001.100/6 | 556  |
| 6 | 281401_4.0001.100/7 | 555  |
| 7 | 281401_4.0001.100/8 | 551  |
| 8 | 281401_4.0001.101/1 | 547  |

Pokaż wszystkie obiekty

Współrzędne: 663216 605659 Skala: 1:11801 Powiększenie: 100% Obrót: 0.0° Renderuj EPSG:2180

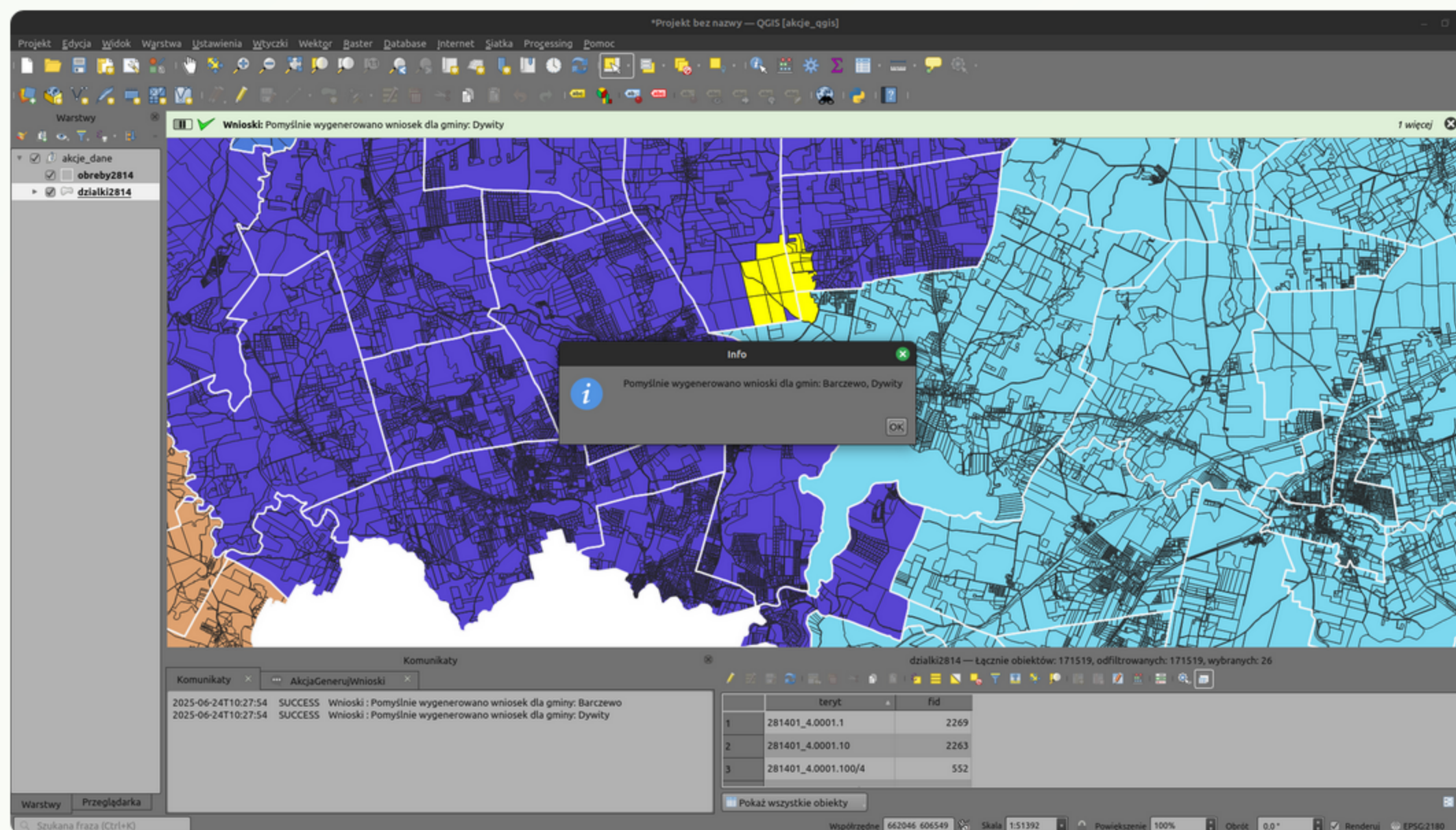
22 / 27



# Akcja generująca wnioski do urzędów gmin



## Okno QGIS po wykonaniu akcji



23 / 27

# Akcja generująca wnioski do urzędów gmin



## Wynikowe pliki \*.txt

```
akcje > wniosek_Barczewo_25.06.2025.txt
1      Łódź, 25.06.2025 r.
2
3
4
5      Urząd Miejski Barczewo
6      pl. Ratuszowy 1
7      11-010 Barczewo
8
9
10     WNIOSEK
11
12
13     Zwracam się z wnioskiem o udostępnienie materiałów dotyczących działki
14     3070/2 w obrębie Barczewko gmina Barczewo powiat olsztyński. Dane te są
15     niezbędne do wykonania niezwykle istotnej mapy w QGIS.
16     Proszę o pozytywne rozpatrzenie mojego wniosku.
17
18
19     Z poważaniem,
20     Tomasz Rychlicki
21
```

```
akcje > wniosek_Dywity_25.06.2025.txt
1      Łódź, 24.06.2025 r.
2
3
4
5      Urząd Gminy Dywity
6      ul. Olsztyńska 32
7      11-001 Dywity
8
9
10     WNIOSEK
11
12
13     Zwracam się z wnioskiem o udostępnienie materiałów dotyczących działek
14     3053/2, 3053/1, 3054/1, 3054/2, 3052, 3042/2, 3072/1, 3072/2, 3073, 180/3,
15     3071/1 w obrębie Dąbrówka Wielka i 3042/1, 206/3, 206/2, 186/12, 203/7,
16     186/3, 203/8, 203/9, 3051/3, 186/6, 186/11, 3051/1, 208, 207 w obrębie Gady
17     gmina Dywity powiat olsztyński. Dane te są niezbędne do wykonania niezwykle
18     istotnej mapy w QGIS.
19     Proszę o pozytywne rozpatrzenie mojego wniosku.
20
21
22     Z poważaniem,
23     Tomasz Rychlicki
24
```

## Stan plików w folderze po uruchomieniu akcji

| /home/tomasz/Pulpit/akcje   |      |         |
|-----------------------------|------|---------|
| Nazwa                       | Roz. | Rozmiar |
| ^ [..]                      |      | <DIR>   |
| wniosek_Barczewo_25.06.2025 | txt  | 765     |
| wniosek_Dywity_25.06.2025   | txt  | 971     |
| akcje_dane                  | gpkg | 58.90 M |
| akcja_wniosek               | py   | 15.52 K |
| wniosek                     | txt  | 671     |



## Najlepsze praktyki

- 1 Używaj opisowych nazw akcji
- 2 Dodawaj komentarze do złożonych skryptów
- 3 Testuj akcje na małych zbiorach danych
- 4 Używaj obsługi błędów (try-except)
- 5 Dokumentuj parametry i wymagania
- 6 Zachowuj kopie zapasowe projektów z akcjami

**Pamiętaj:** dobrze napisana akcja to akcja zrozumiała i wielokrotnego użytku





## Praktyczne przypadki użycia

- **Inspekcje terenowe:** przejście do Street View lub Geoportalu
- **Analiza historyczna:** linki do zdjęć archiwalnych
- **Raportowanie:** generowanie PDF dla obiektów
- **Integracja z bazami:** aktualizacja zewnętrznych systemów
- **Walidacja danych:** sprawdzanie poprawności geometrii
- **Eksport selektywny:** zapis wybranych obiektów

Akcje idealnie nadają się do automatyzacji rutynowych zadań



# Dziękuję za uwagę!