

## ZAAWANSOWANE JS W PRZEGŁĄDARCE - PUZZLE

### SPIS TREŚCI

Spis treści .....	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga .....	1
Wymagania.....	2
Implementacja .....	2
Commit projektu do GIT.....	9
Podsumowanie.....	9

### CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- pobieranie lokalizacji w przeglądarce z wykorzystaniem Geolocation API
- wyświetlanie map z wykorzystaniem biblioteki Leaflet
- pobieranie map statycznych z wykorzystaniem Leaflet
- podział obrazów na sekcje z wykorzystaniem JS
- przestawianie elementów z wykorzystaniem Drag & Drop
- wyświetlanie powiadomień

W praktycznym wymiarze uczestnicy stworzą dynamiczną aplikację – układankę – w której gracz będzie musiał ułożyć 16 elementów uprzednio wskazanej i pobranej mapy.

### ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtóżenie Geolocation API, Leaflet, Drag & Drop, Canvas.

Wejściówka?

### UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do Plik -> Informacje -> Właściwości -> Właściwości zaawansowane -> Niestandardowe i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub Ctrl+A -> F9.

## WYMAGANIA

W ramach LAB C przygotowane powinny zostać:

- pojedyncza strona HTML ze skryptem ładowanym z zewnętrznego pliku JS
- pobranie zgody na lokalizację
- pobranie zgody na wyświetlanie powiadomień
- okno dynamicznej mapy (powiększanie/pomniejszanie, przesuwanie)
- przycisk „Moja lokalizacja” – wyświetla współrzędne oraz oznacza na mapie
- przycisk „Pobierz mapę” – eksportuje mapę w postaci rastrowej
- mapa rastrowa zostaje podzielona na 16 elementów i wymieszana; elementy rozrzucone na „stole”
- użycie mechanizmu drag & drop do przemieszczania elementów na „stole”
- w tle weryfikacja czy element ustawiony na swoim miejscu
- w momencie ustawienia wszystkich elementów na swoim miejscu – wyświetlenie notyfikacji

Wideo z omówieniem działającej aplikacji: [https://www.youtube.com/watch?v=Peb\\_mgDTY0s](https://www.youtube.com/watch?v=Peb_mgDTY0s).

Prowadzący omówi powyższe wymagania. Upewnij się, czy wszystko rozumiesz.

Tu umieść swoje notatki:

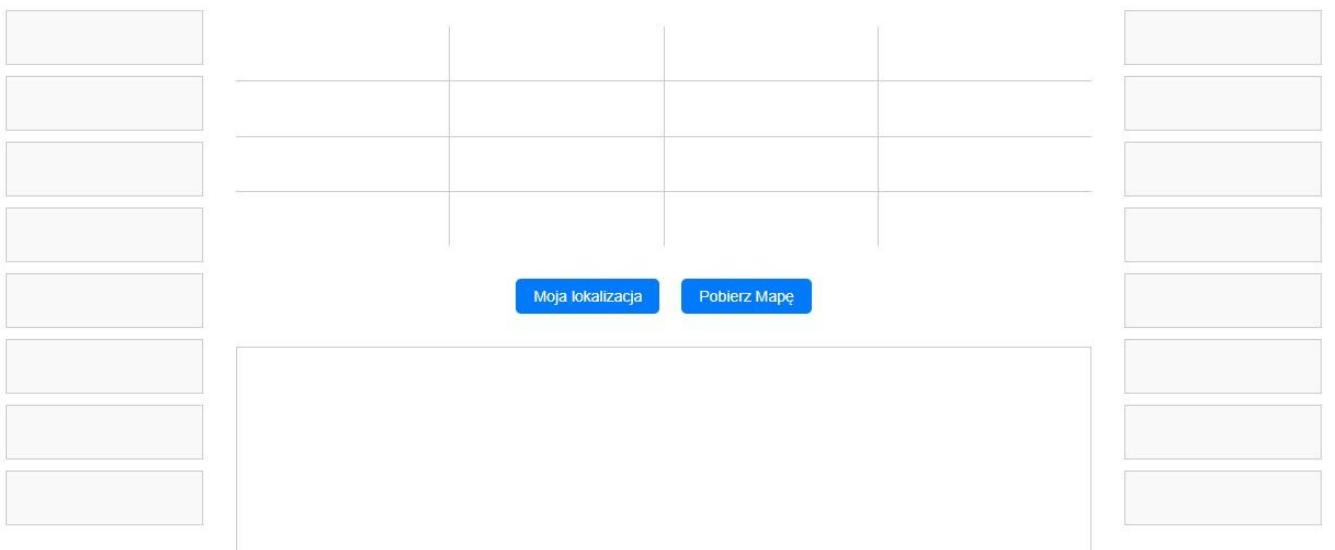
...notatki...

## IMPLEMENTACJA

Tradycyjnie implementację należy zacząć od zbudowania w HTML + CSS wszystkich wymaganych elementów / placeholderów na te elementy. Następnie krok po kroku należy implementować poszczególne zachowania.

**UWAGA!** Większość kodu jest już zrobiona! Wystarczy przejrzeć kody źródłowe prezentowane na wykładach 😊

Wstaw zrzut ekranu zawierającego stronę ze wszystkimi elementami, tj. okno dynamicznej mapy, przycisk „Moja lokalizacja”, przycisk „Pobierz mapę”, przestrzeń z rozsypanymi puzzlami, przestrzeń do układania puzzli. Wygląd może być odmienny od zaprezentowanego na wideo:

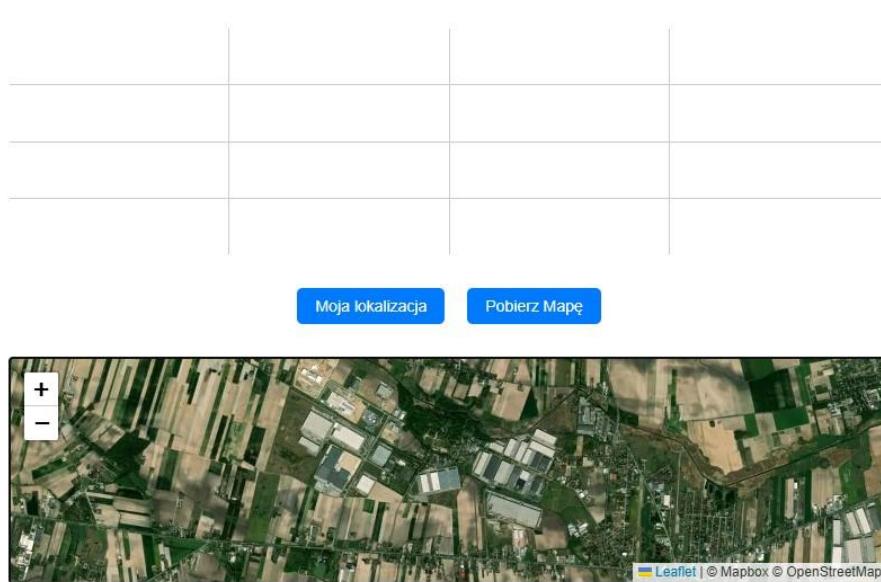
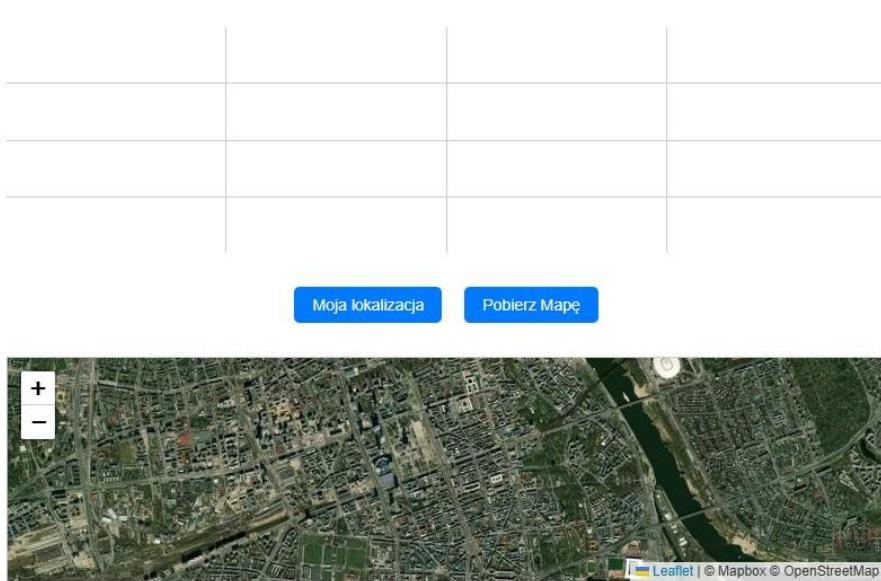


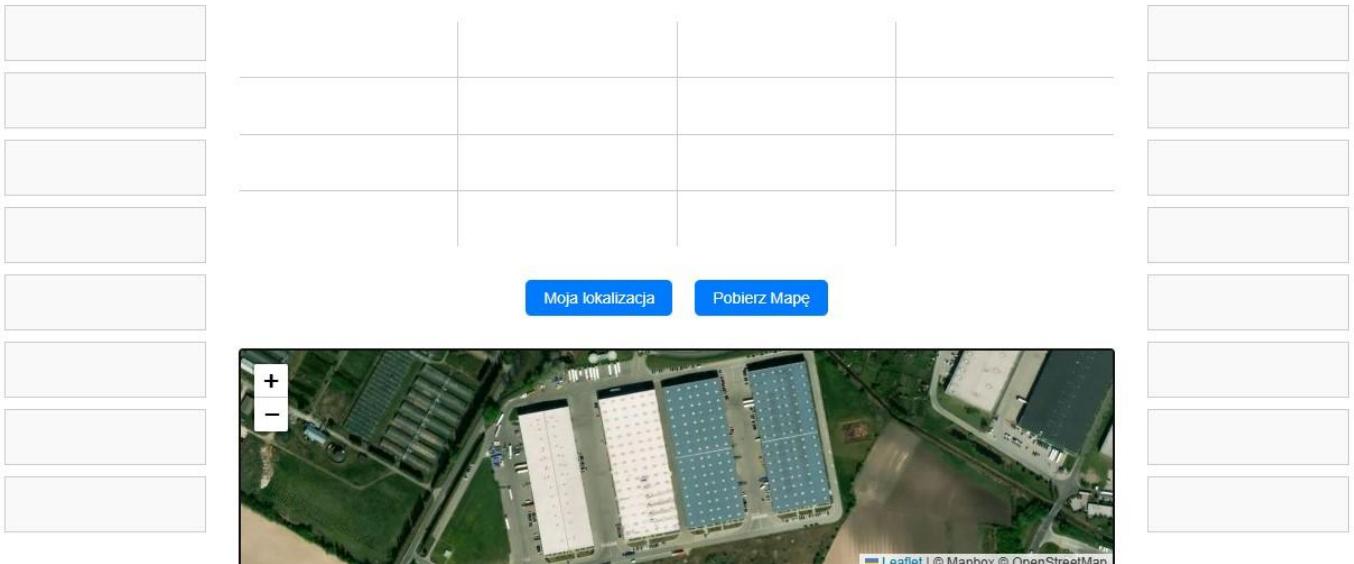
Punkty:

0

1

Wstaw zrzuty ekranu z załadowaną dynamiczną mapą, inną lokalizacją na mapie, innym przybliżeniem mapy:



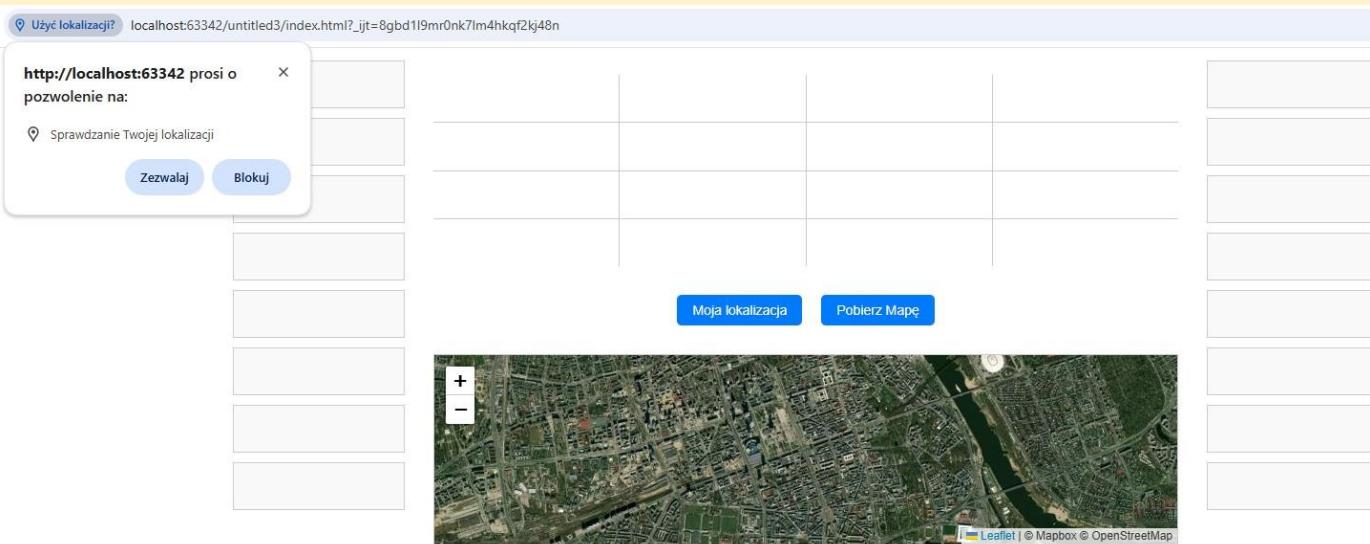


Punkty:

0

1

Przedstaw zrzut ekranu przeglądarki proszącej o zgodę na udostępnienie geolokalizacji:



Wstaw zrzut ekranu wycentrowanej mapki na pobranej geolokalizacji:


[Moja lokalizacja](#) [Pobierz Mapę](#)



Punkty:

0	1
---	---

Wstaw zrzut ekranu obrazujący zapisanie ustawionego fragmentu mapy dynamicznej do rastra w canvas:


[Moja lokalizacja](#) [Pobierz Mapę](#)




Punkty:

0	1
---	---

Wstaw zrzut ekranu obrazujący podział mapy rastrowej na puzzle:



Punkty:

0

1

Wstaw dwa zrzuty ekranu obrazujące działanie mechanizmu Drag-and-Drop na puzzlach:

A 4x4 grid puzzle interface. The central area contains a satellite map of a city with a river, flanked by zoom controls (+ and -). Below the map are two blue buttons: "Moja lokalizacja" and "Pobierz Mapę". The grid itself consists of 16 smaller images. The top row contains three white squares and one satellite image. The second row contains two satellite images and two white squares. The third row contains two white squares and two satellite images. The bottom row contains three satellite images and one white square.

A 4x4 grid puzzle interface, similar to the one above, showing a partially solved state. The central map and zoom controls are identical. The grid contains 16 images. The top row has three white squares and one satellite image. The second row has two satellite images and two white squares. The third row has two white squares and two satellite images. The bottom row has three satellite images and one white square. Some images in the grid appear to be partially overlapping or misaligned, indicating they have been moved but not yet correctly placed.

Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu obrazujący działający mechanizm wykrywania poprawnego ułożenia wszystkich puzzli. Można ograniczyć się do wydrukowania komunikatu za pomocą `console.log()`:

DevTools is now available in Polish! [Always match Chrome's language](#) [Switch DevTools to Polish](#) [Don't show again](#) X

Elements **Console** Sources Network Performance Memory Application > ⚙️ ⋮ X

top ▾ Filter Default levels ▾ No Issues ⚙️

▶ 1 message app.js:240  
▶ 1 user mes...  
✖ No errors  
⚠ No warnings  
▼ 1 info  
📄 app.js 1  
⚙️ No verbose

Gratulacje, puzzle zostały poprawnie ułożone!

Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu obrazujący wyświetlenie notyfikacji systemowej po poprawnym ułożeniu puzzli:

Leaflet © Mapbox © OpenStreetMap

Punkty:	0	1
---------	---	---

## COMMIT PROJEKTU DO GIT

Zacommituj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie `lab-c` na podstawie głównej gałęzi kodu.

Podaj link do brancha `lab-b` w swoim repozytorium:

...link, <https://github.com/mm51621/main/tree/main/AI/labC>

## PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Tworzenie elementów z drag and drop

Łączenie z zewnętrznym API

Lokalizacja użytkownika z wykorzystaniem Geolocation API

Tworzenie powiadomień systemowych

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.