|  |  |  |
| --- | --- | --- |
| **AI1** | Nazwisko Imie  Album 00000  Grupa 1 | **Wersja 1** |
|  |
| **LAB C** |

Zaawansowane JS w przeglądarce - puzzle

# Spis treści

[Spis treści 1](#_Toc147182537)

[Cel zajęć 1](#_Toc147182538)

[Rozpoczęcie 1](#_Toc147182539)

[Uwaga 1](#_Toc147182540)

[Wymagania 2](#_Toc147182541)

[Implementacja 2](#_Toc147182542)

[Commit projektu do GIT 5](#_Toc147182543)

[Podsumowanie 5](#_Toc147182544)

# 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 – puzzle, w której gracz będzie musiał ułożyć 16 elementów uprzednio wskazanej i pobranej mapy.

# Rozpoczęcie

Rozpoczęcie zajęć. Powtórzenie 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 omówienie działającej aplikacji: <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:



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

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



|  |  |  |
| --- | --- | --- |
| 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:

 

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

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



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

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



|  |  |  |
| --- | --- | --- |
| 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, np. <https://github.com/inazwisko/ai1-lab/tree/lab-c…>

# Podsumowanie

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

…podsumowanie…

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.