

Laboratorium Systemów Informacji Przestrzennej

Tworzenie internetowego Systemu Informacji Przestrzennej przy użyciu OpenLayers

1. Wprowadzenie do API biblioteki OpenLayers

Biblioteka OpenLayers jest darmowym narzędziem pozwalającym na tworzenie sieciowych systemów informacji przestrzennej które funkcjonalnością niewiele ustępują produktom firmy Google. Ponieważ biblioteka ta jest rozwiązaniem typu Open Source, możliwe jest rozszerzanie jej istniejących możliwości o nowe elementy jak również całkowita zmiana sposobu jej działania.

Z podanego przez prowadzącego adresu należy pobrać plik **OpenLayersMap.html** i zapisać na dysku, a następnie otworzyć np. w Notatniku. Pierwszym rzucającym się w oczy elementem tego pliku jest lista importowanych skryptów, znajdująca się zaraz za znacznikiem `</title>`. Ponieważ OpenLayers nie jest powiązane z żadną konkretną firmą, pozwala na pobieranie i wyświetlanie danych oraz map z dowolnego źródła. W bieżącym dokumencie wykorzystano metody pozwalające na obsługę warstw z serwisów Bing Maps oraz OpenStreetMap. Kolejnymi elementami skryptu strony są deklaracje zmiennych `lon`, `lat` oraz `zoom`, które można wykorzystać podczas wykonywania ćwiczenia, zmienna `map` reprezentująca obiekt mapy, obiekty zawierające definicje projekcji Mercator World oraz WGS84, przykładowy klucz `api` Microsoft Bing oraz obiekt `options` zawierający początkowe ustawienia mapy (w tym przypadku jest to tylko ustawienie domyślnej projekcji i układu współrzędnych na wartość Mercator World).

Za utworzenie mapy oraz elementów interfejsu użytkownika w OpenLayers odpowiada funkcja `init()`. Konstruktor `OpenLayers.Map('map')` tworzy nowy obiekt typu `Map` wewnątrz podanego kontenera HTML (w tym przypadku obiektu `div` o nazwie `map`). Metoda `AddLayers([...])` dodaje do mapy tablicę warstw (w danym przypadku tablica zawiera tylko jedną warstwę), natomiast metoda `zoomIn` powoduje pojedyncze przybliżenie widoku mapy.

2. Wykorzystywanie wbudowanych metod podstawowych obiektów OpenLayers

Początkowo strona ma tylko jedną warstwę bazową, która zawiera dane z serwisu OpenStreetMap. W podobny sposób należy dodać do mapy dane z serwisu Bing Maps. W celu stworzenia warstwy zawierającej mapę ulic należy stworzyć nową zmienną, np. o nazwie `bing_streets` i przypisać do niej nowy obiekt typu `OpenLayers.Layer.Bing`. W tym przypadku konstruktor ma postać:

```
OpenLayers.Layer.Bing({  
  name: "nazwa",  
  key: apiKey,  
  type: "typ"  
}),
```

gdzie `apiKey` to zmienna przechowująca klucz wygenerowany przy pomocy strony <http://bingmapsportal.com>, zaś zmienna `type` może przyjmować wartości:

- `type: "Road"`

dla warstwy ulic,

- `type: "Aerial"` dla warstwy satelitarnej, oraz
- `type: "AerialWithLabels"`

dla warstwy hybrydowej.

Po utworzeniu potrzebnych obiektów, należy je dodać do listy wyświetlanych warstw. W tym celu należy do istniejącego polecenia `map.addLayers([osm])` dodać utworzone przez siebie obiekty. Przykładowo w przypadku warstw nazwanych `bing_road`, `bing_phys` oraz `bing_sat` instrukcja powinna mieć postać `map.addLayers([osm, bing_road, bing_phys, bing_sat]);`.

Domyślnie mapa zawiera kontrolkę pozwalającą na proste operacje typu przybliżanie/oddalanie i przesuwanie widoku. Należy dodać do mapy nowe narzędzie typu `OpenLayers.Control.LayerSwitcher()`, pozwalające na wybór wyświetlanych warstw. Służy do tego metoda `map.addControl()`. Przy jej pomocy należy dodać do mapy i przetestować funkcjonalność następujących kontrolek:

- `OpenLayers.Control.PanZoomBar()`, pozwalającą na zaawansowaną kontrolę skali mapy,
- `OpenLayers.Control.ScaleLine()`, wyświetlającą aktualną skalę mapy,
- `OpenLayers.Control.OverviewMap()`, wyświetlająca w rogu ekranu pomniejszoną mapę okolicy,
- `OpenLayers.Control.KeyboardDefaults()`, pozwalającą na poruszanie się po mapie przy pomocy klawiatury,
- `OpenLayers.Control.MousePosition()`, podającą aktualne współrzędne kursora myszy.

Po dodaniu do mapy powyższych elementów należy zmienić rodzaj podawanych przez narzędzie `MousePosition` współrzędnych z metrów Mercatora na długość i szerokość geograficzną. W tym celu należy do pola `displayProjection` obiektu `map` przypisać właściwy obiekt reprezentujący definicję projekcji WGS84.

Następnie należy umieścić w pliku kod powodujący wycentrowanie widoku mapy na punkcie wskazującym skrzyżowanie ulic Okopowej i Podwałe Przedmiejskie w okolicy centrum Gdańska (współrzędne: 18.64542, 54.34766). W tym celu należy wykorzystać następujący kod: `map.setCenter(new OpenLayers.LonLat(długość, szerokość), zoom);`, gdzie wartość `zoom` jest liczbą odpowiadającą konkretnemu poziomowi przybliżenia mapy (zakładając, iż poziom „0” oznacza widok całego świata). Na początek parametrowi `zoom` należy nadać wartość „5”. Ponieważ układ współrzędnych mapy operuje na metrach Mercatora, bezpośrednie podanie parametrów w postaci długości i szerokości geograficznej nie pozwoli osiągnąć pożądanego rezultatu. Aby podane współrzędne zostały odpowiednio zinterpretowane, należy dokonać ich transformacji z układu WGS84 do World Mercator za pomocą metody obiektu `LonLat` o nazwie `transform(ukł_pocz, ukł_docelowy)`.

Następnie należy dodać warstwę WMS typu `overlay`, tzn. taką, która może być nałożona na dowolną z warstw bazowych. W tym celu wykorzystamy zdjęcie pochodzące z będącej w posiadaniu KSG stacji satelitarnej. Należy utworzyć nową warstwę `OpenLayers.Layer.WMS()` o parametrach:

- `"nazwa"`,
- `"adres_serwera"`, czyli `"http://ksg.eti.pg.gda.pl/geoserver/gis/wms"`,
- `{layers: '2014_07_25_Baltic_Cyano', transparent: true}, {isBaseLayer: false}` - na tej liście parametrów zaawansowanych pierwszy pozwala na grupowanie kilku podwarstw w ramach jednej "fizycznej" warstwy, drugi zaś pozwala określić, czy dodana warstwa ma być warstwą bazową (domyślnie), czy nakładaną na warstwę bazową (czyli interesującego nas typu `overlay`).

i dodać ją do mapy poleceniem `map.addLayer(...)`. Następnie należy sprawić aby nowo dodana warstwa stała się półprzezroczysta. W tym celu należy wykorzystać standardową dla wszystkich warstw funkcję `setOpacity(...)`, która jako parametr przyjmuje wartość z zakresu {0...1}.

Kolejnym elementem ćwiczenia jest stworzenie warstwy wektorowej pozwalającej na tworzenie własnych kształtów. W tym celu należy stworzyć nową zmienną, np. o nazwie `"vectors"`. Do tej zmiennej należy przypisać nową warstwę typu `OpenLayers.Layer.Vector("nazwa")` po czym należy dodać ją do mapy. Następnie należy dodać do mapy nową kontrolkę typu `OpenLayers.Control.EditingToolbar()`, której należy w postaci parametru przekazać utworzoną

uprzednio zmienną *vectors*. Na nowo dodanej warstwie wektorowej należy stworzyć po trzy obiekty typu *point*, *line* i *polygon*.

Na zakończenie ćwiczenia należy dodać do mapy warstwę zawierającą markery wskazujące trzy wybrane miasta. W tym celu należy stworzyć nową warstwę typu *OpenLayers.Layer.Markers("nazwa")* i dodać ją do mapy. Następnie należy stworzyć trzy nowe obiekty typu *OpenLayers.LonLat()* zawierające potrzebne współrzędne. Nowe markery należy utworzyć przy pomocy konstruktora *OpenLayers.Marker()* którego parametrem powinien być odpowiedni obiekt typu *LonLat*. W celu dodania markerów do warstwy można wykorzystać metodę *addMarker()* warstwy markerów.

ZADANIA DO WYKONANIA

1. Dodanie do mapy wszystkich warstw pochodzących z serwisu Microsoft Bing. 1 pkt.
2. Dodanie do mapy wszystkich wymienionych w instrukcji kontrolek 1 pkt.
3. Dodanie do mapy półprzezroczystej warstwy WMS 1 pkt.
4. Dodanie do mapy warstwy wektorowej i narysowanie na niej opisanych w instrukcji obiektów 1 pkt.
5. Dodanie do mapy warstwy markerów zawierającej trzy wybrane miasta 1 pkt.