

Darbas su *Google Maps API*

Laboratorinio darbo vertinimas: šio darbo vertė sudaro 7 proc. viso kurso bendro pažymio.

APRAŠYMAS IR UŽDAVINIAI

Šiame laboratoriniame darbe norėdami pakeisti geografinę projekciją (ilgumą ir platumą), į *ArcMap* įkelsime duomenis apie Vilniaus greitosios pagalbos stotis. Antroje dalyje sukursime *JavaScript* funkciją, skirtą *Google Map* rodyti greitosios pagalbos stočių pavadinimus ir vietas naudojant *Google Maps* programų sąsajos (*Application Programming Interface – API*) funkcijas.

Baigę šį darbą, turėtumėte suprasti šiuos terminus ir sąvokas:

- API
- *Google Maps API*
- Programa duomenims iš įvairių šaltinių jungti (*Mashup*)
- *JavaScript*

ATSISKAITYMO REIKALAVIMAI

Atlikdami šią užduotį, sukursite svetainę. Baigę dėstytojui pateikite **index.html** tinklalapio adresą. Jūsų svetainėje kartu su *Google* žemėlapiu turi būti panaudoti greitosios pagalbos stočių duomenys. Vertinant bus atsižvelgta į reikalaujamo derinio buvimą.

Ivadas

Viena unikali *Google Maps* funkcija yra laisvai prieinamas API kodas, leidžiantis vartotojams naudoti *Google Map* duomenis išorinėse svetainėse, tinklaraščiuose ir kituose tinklalapiuose. *Google* sukūrė ***Google Maps API*** norėdama palengvinti *Google Maps* integraciją svetainėse su nuosavomis duomenų saugyklomis. Ši technologija yra paplitusi visame pasaulyje. Šiuo metu ***Google Maps API*** yra nemokama paslauga be reklamos, nors *Google* ir pasiliko teisę *Google Maps API* rodyti reklamą ateityje.

Google Maps API yra viena iš daugelio *Google* teikiamų vartotojo sąsajų. Ji apima funkcijas, skirtas žemėlapiu braižymui, centravimui, mastelio keitimui ir polilinijų bei žymeklių braižymui žemėlapiu viršuje. Paprastą pavyzdį žr. <http://tree.discovery.mala.bc.ca/google>.

Norėdamas pradėti naudoti *Google Maps API*, vartotojas turi sukurti API raktą. Šis raktas ribos priėjimą prie svetainės ir katalogo, kuriuos vartotojas nurodys kurdamas raktą. Nuosavos žemėlapiu sąsajos kūrimas apima *Google JavaScript* kodo įrašymą į vartotojo tinklalapį, o tada *JavaScript* funkcijų naudojimą norint sujungti vartotojo turinį su žemėlapiu. Kai pradėjo veikti API, jai trūko galimybės geokoduoti adresus, todėl vartotojai taškus (ilgumą ir platumą) turėjo nurodyti rankiniu būdu. Vėliau šis trūkumas buvo ištaisytas.


Tuo pačiu metu, kai pasirodė *Google Maps API*, *Yahoo!* išleido savo *Maps API*. *Yahoo! Maps* trūko tarptautinio palaikymo, tačiau jau pirmojoje versijoje buvo geokodavimo įrankis.


Kaip ir *Google* atveju, *Maps API* yra tobulinama, todėl sąsajoje bet kuriuo metu gali būti atlikti pakeitimai. Šis laboratorinis darbas buvo atnaujintas, kad suteiktų informacijos, kaip naudotis esama (v=2) *Google Maps API* versija.

1 etapas: paruoškite duomenis

1. Nukopijuokite aplanką *GII-09 Lab4 Data* (CD nr.8/Duomenys/GII-09 Lab4 Data) su 4 laboratorinio darbo duomenimis į darbinį aplanką savo kompiuteryje.

Google žemėlapių jungimas veikia tik su ilgumos ir platumos koordinatėmis. Todėl *Google KLM* palaiko geografines koordinates tik WGS84 atskaitos sistemoje ir aukštį kaip atstumą virš Žemės paviršiaus (metrais). *Google KML (Keyhole Markup Language)* yra XML pagrįsta žymėjimo kalba, skirta *Google Maps* elektroninių žemėlapių paslaugai ir papildomiems vartotojo nurodytoms elementams rodyti *Google Maps* sistemoje. Šiuo atveju, jei norima sujungti duomenis *Google Map*, reikia paruošti greitosios pagalbos stočių vietas geografinėje koordinacinių sistemoje.

2. Paleiskite *ArcMap* ir iš šio darbinio aplanko įkelkite failą **Ambulance stations.shp**. Naudodami *ArcToolbox* , pakeiskite projekciją iš *Lietuvos_Koordinacių_Sistema* į *Geographic* (pasirinkite *Data Management Tools* (duomenų valdymo įrankiai) > *Projection and Transformations* (projekcija ir transformacijos) > *Feature* (elementas) > paleiskite *Project* (projektuoti) > failą **Ambulance stations.shp** pakeiskite į **Ambulance_stations.shp** (su brūkšneliu) ir projektuokite į *World Geographic WGS 1984*.
3. Prie **Ambulance_stations** failo atributų lentelės pridėkite laukus **POINT_X** ir **POINT_Y**. Dešiniuoju mygtuku spustelėkite slauksnį **Ambulance_stations** > *Open Attribute Table* (atverti atributų lentelę) > *Options* (parinktys) > pirmiausia pridėkite lauką (*Add Field*), pavadintą **POINT_X**, įveskite *Double* (dvigubas), *pension 15* ir *scale 10* > *OK* (gerai). Dešiniuoju mygtuku spustelėkite **POINT_X** lauko antraštę > *Calculate Geometry* (apskaičiuoti geometriją) ir pasirinkite *X Coordinate for Point, Use coordinate system of the data source* (X koordinatė taškui, naudoti duomenų šaltinio koordinacinių sistemą) t.y. *GCS: WGS 1984* ir vienetus *Decimal Digress* (dešimtainis nuokrypis) > *OK* (gerai).
4. Sukurkite lauką **POINT_Y** ir taškinei geometrijai apskaičiuokite Y koordinates.
5. Dabar eksportuokite **Ambulance_stations** atributų lentelę į tekstinį failą: *Options* (parinktys) > eksportuokite (*Export*) visus įrašus į išvesties lentelę **Ambulance_stations.txt** darbiniam aplanke.

Eksportuojamų duomenų tipą į **.txt** pakeisti galite duomenų išsaugojimo dialogo lange . Spustelėkite *Save* (išsaugoti) ir *OK* (gerai). Kad prie esamo žemėlapio nebūtų pridėta nauja tekstinė lentelė, pasirinkite *No* (ne).

6. Tekstų redaktoriais, pvz., *NotePad*, atidarykite **Ambulance_stations.txt** atributų lentelę ir ištrinkite visus lentelės stulpelius, išskyrus "**ADRESAS**", "**POINT_X**" ir "**POINT_Y**" bei kitus, kuriuos vėliau norėsite matyti savo žemėlapyje. Antraštė ir pora kableliais atskirtų **Ambulance_stations.txt** įrašų gali atrodyti taip:

```
"ADRESAS", "POINT_X", "POINT_Y"  
"Pamėnkalnio g. 30", 25.274405, 54.685821  
"R. Jankausko g. 2/28", 25.208704, 54.689791  
"Dvitrigailos g. 18", 25.268299, 54.676133  
...
```

Eksportavote greitosios pagalbos stotis ir X, Y koordinates į formatą, kuris gali būti naudojamas su *JavaScript* programa. Dabar galima sukurti funkciją, nuskaitančią šiuos duomenis (su nedideliais pakeitimais).

2 etapas: užregistruokite ir sukurkite *Google* paskyrą

Norėdami naudotis *Google Maps API*, turite užsiregistruoti *Google* sistemoje ir susikurti *Google* paskyrą. Užsiregistravę *Google*, turite nurodyti absoliutųjį adresą (*Uniform Resource Locator* – URL), kur saugomi jūsų tinklalapiai. Naudojantys *Google Maps API* tinklalapiai turi būti pasiekiami šiuo adresu ir tam, kad veiktų, turi turėti kode nurodytą raktą.

1. Interneto naršyklėje įveskite <http://code.google.com/apis/maps/> > norėdami gauti raktą, užsiregistruokite šiame tinklalapyje ([Sign up for a Google Maps API key](#)) > perskaitykite ir sutikite su nuostatomis ir sąlygomis, įveskite svetainės, kurioje bus jūsų *Google Map* žemėlapis, adresą (pvz., <http://www.geocities.com/yourname/>).
2. Nukopijuokite ir išsaugokite jūsų *Google Maps API* raktą.

3 etapas: sujunkite miesto vietas su *Google Map*

1. Prieš pridėdami vietas, atlikime paprasčiausią užduotį, kad patikrintumėme, ar viskas veikia. *Google Maps API* pradžios tinklalapį žr. <http://www.google.com/apis/maps/>, dokumentaciją žr. <http://www.google.com/apis/maps/documentation/>. Panagrinėkime pavyzdį. Norėdami matyti Vilnių, galime pakeisti *Google Maps* „Hello World“ pavyzdį.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>  
    <title>Google Maps JavaScript API Example</title>  
    <script src="http://maps.google.com/maps?file=api&v=2&key=abcdefg"  
      type="text/javascript"></script>  
    <script type="text/javascript">  
      //<![CDATA[  
  
      function load() {  
        if (GBrowserIsCompatible()) {  
          var map = new GMap2(document.getElementById("map"));  
          map.setCenter(new GLatLng(37.4419, -122.1419), 13);  
          map.openInfoWindow(map.getCenter(),  
            document.createTextNode("Hello, world"));  
        }  
      }  
    ]></script>  
</html>
```

```

//]]>
</script>
</head>
<body onload="load()" onunload="GUnload()">
<div id="map" style="width: 500px; height: 300px"></div>
</body>
</html>

```

Iš tiesų šiame kode yra tik trys svarbios eilutės – tai eilutės prasidedančios `if (GBrowserIsCompatible())`. Programai veikiant jums tereikia pakeisti šį kodą, kad būtų atliekami jūsų pageidaujami veiksmai.

2. Nukopijuokite šį pavyzdį į tekstų redaktorių (*Notepad* ar kitą HTML redaktorių) ir pakeiskite 7 eilutėje esantį raktą iš „abcdefg“ į jūsų *Google Maps API* raktą. Išsaugokite failą pavadinimu **GoogleMap.html**.

Pastaba: teksto neveskite rankomis. Nukopijuokite iš šio dokumento ir įkelkite į tekstų redaktorių.

3. Dabar pakeiskite ilgumos ir platumos reikšmes, kad žemėlapyje būtų rodomas Vilnius. Vilniaus koordinatės yra 54,696 laipsniai šiaurės platumos ir 25,278 laipsniai rytų ilgumos. Pakeiskite tekstą „Hello, world“ į „Hello, Vilnius“.
4. Dabar metas išbandyti veikimą. Įkelkite **GoogleMap.html** failą į savo svetainę (pvz., <http://www.geocities.com/yourname/>) ir peržiūrėkite **GoogleMap.html** tinklalapį.

Pastaba: keičiant kurį nors kodą, geriausia atlikti tik vieną ar du pakeitimus ir tada išbandyti, kaip šie pakeitimai veikia. Pakeitimus galite atlikti tiesiog faile, naudodami svetainės HTML redaktorių, arba juos atlikite kompiuteryje esančiame faile, o tada tinklalapį įkelkite į svetainę.

4 etapas: sujunkite visas Vilniaus greitosios pagalbos stočių vietas su *Google Map*

1. Kadangi paprastas žemėlapis rodomas, metas išbandyti šį tą sudėtingesnio. Pirmiausia išnagrinėkime *Google Maps API* dokumentacijos pavyzdį "*Display Info Windows Above Markers*". Čia pateiktas pavyzdžio kodas su nedideliais pakeitimais ir pridėtais komentarais:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Google Maps JavaScript API Example</title>
<script
src="http://maps.google.com/maps?file=api&v=2&key=your_Google_Maps_API_key"
-->
type="text/javascript"></script>
<script type="text/javascript">
//

function createMarker(X, Y, number) { // This is the function to modify
var point = new GLatLng(X,Y);
var marker = new GMarker(point);
// Show this marker's index in the info window when it is clicked
var html = "Marker #&lt;b&gt;" + number + "&lt;/b&gt;"; // HTML Code goes in here.
GEvent.addListener(marker, "click", function() {
marker.openInfoWindowHtml(html);
});
</pre>
</div>
<div data-bbox="851 945 919 960" data-label="Page-Footer">
<p>4 iš 6 psl.</p>
</div>
```

```

return marker; // Rather than returning the marker, we can just add it here
}

function load() {
if (GBrowserIsCompatible()) {
var map = new GMap2(document.getElementById("map")); // Set up map
map.setCenter(new GLatLng(37.4419, -122.1419), 13); // Center the map on Palo
Alto 13 is the zoom level. Higher numbers zoom in
map.addControl(new GSmallMapControl()); // Add controls
map.addControl(new GMapTypeControl());

// Add 10 markers in random locations on the map
// Most of this function is not needed
var bounds = map.getBounds(); // We will just add the markers at the X, Y
coordinates of the locations
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i=0; i<10; i++) {
var X=southWest.lat() + latSpan * Math.random();
var Y=southWest.lng() + lngSpan * Math.random();

map.addOverlay(createMarker(X, Y, i)); //map.AddOverlay can be incorporated
into createMarker
}
}
}

//]]>
</script>
</head>
<body>
<h3><center>Vilnius Ambulance Stations </h3>
<body onload="load()" onunload="GUnload()"> <!-- the body doesn't really do
anything except position the map -->
<div id="map" style="width: 500px; height: 300px"></div>
</body>
</html>

```

2. Dabar, pasinaudodami šiuo pavyzdžiu, sukurkite funkciją, nuskaitančią **ADRESAS**, **POINT_X** ir **POINT_Y** formos duomenis ir sukuriančią žymeklius X, Y vietose kartu su išskylančiais langais, kuriuose būtų rodomi greitosios pagalbos stočių pavadinimai. Kitaip sakant, funkcija turėtų atrodyti taip:

```
createMarker (ADRESAS, POINT_X, POINT_Y)
```

3. Norėdami tai pasiekti, atlikite nurodytus veiksmus, kiekvieną žingsnį tikrindami gaunamus rezultatus:
 - Naudodami tekstų redaktorių (*Notepad* ar kitą HTML redaktorių) atidarykite pirmojo laboratorinio darbo failą **JavaScript.html** ir ištrinkite visą kodą
 - Nukopijuokite "*Display Info Windows Above Markers*" pavyzdį į **JavaScript.html** ir išsaugokite šį failą
 - Pakeiskite raktą iš *your_Google_Maps_API_key* į jūsų *Google Maps API* raktą
 - Pakeiskite *CreateMarker* argumentus į **ADRESAS**, **POINT_X** ir **POINT_Y**. Argumentams nenaudokite kabučių
 - Pakeiskite *GLatLng* argumentus į **POINT_X** ir **POINT_Y**. Atkreipkite dėmesį, kad pirmasis šios funkcijos argumentas yra plotuma, t. y. **POINT_Y**, o antrasis argumentas yra ilguma, t. y. **POINT_X**

- Argumentui ADRESAS priskirkite var html, pvz., var html = ADRESAS";
- Pakeiskite sakinį return marker; į map.addOverlay(marker);
- Pakeiskite GLatLng funkcijos map.setCenter(new GLatLng(49.165866, -123.95557), 13); argumentus į 54.696, 25.278 Pakeiskite mastelio lygį iš 13 į 11
- Ištrinkite nurodytą kodo dalį:

```
// Add 10 markers in random locations on the map
// Most of this function is not needed
var bounds = map.getBounds(); // We will just add the markers at the X, Y
coordinates of the locations
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i=0; i<10; i++) {
var X=southWest.lat() + latSpan * Math.random();
var Y=southWest.lng() + lngSpan * Math.random();

map.addOverlay(createMarker(X, Y, i)); //map.AddOverlay can be incorporated
into createMarker
}
```

- Ištrinkite prieš tai pateiktą kodą ir kiekvienai greitosios pagalbos stočiai pakeiskite atsitiktinius žymeklius į funkcijos createMarker kreipinius, pvz.:

```
map.addOverlay(createMarker("Pamėnkainio g. 30",25.274405,54.685821))
map.addOverlay(createMarker("R.Jankausko g. 2/28",25.208704,54.689791))
```

...

Įrašykite tokį kodą greitosios pagalbos stotims. Galie nukopijuoti failo **Ambulance_stations.txt** tekstą į kodą ir, naudodami iškirpimo ir įkėlimo funkcijas, tekstiniaame faile įrašyti reikiamus funkcijos kreipinius.

- Tarp <body> ir </body> žymų galite įterpti antraštę, pvz., <h3><center>Vilnius Ambulance Stations </h3>

4. Norėdami vietiniame kompiuteryje išbandyti tinklalapį, atidarykite jį interneto naršyklėje.

Pastaba: *JavaScript* nėra gerai pritaikyta redagavimui tekstų redaktoriu. Jei jūsų *JavaScript* kode yra klaida, interneto naršyklėje nematysite vaizdo. Interneto naršyklė pranešimą apie klaidą gali rodyti tik apačioje esančioje būsenos juostoje. Jei jūsų ekranas tuščias, pakartotinai patikrinkite kodą, nukopijuokite iš naujo ir pabandykite dar kartą.

5 etapas: svetainės publikavimas

Paskutiniu šio laboratorinio darbo etapu įkelkite tinklalapį į talpinimo svetainę.

1. Nueikite į jūsų talpinimo svetainę (pvz., <http://www.geocities.com/yourname/>) ir ištrinkite ten esantį seną **JavaScript.html**.
2. Įkelkite **JavaScript.html** failą į savo svetainę (pvz., <http://www.geocities.com/yourname/>) ir peržiūrėkite **JavaScript.html** tinklalapį.

Atlikę visus prieš tai nurodytus veiksmus, galite peržiūrėti savo tinklalapį internete. Dešiniajame rėmelyje turi būti 1 laboratoriniame darbe sukurta nuoroda į **JavaScript.html** su greitosios pagalbos stotimis. Įvertinimui pateikite **index.html** pradžios tinklalapio URL nuorodą (pvz., <http://www.geocities.com/yourname/>).