



# Internetes térképi alkalmazások készítése script nyelvek használatával

Gede Mátvás

ELTE Térképtudományi és Geoinformatikai Tanszék

## 1. Bevezetés

A térképek internetes megjelenítése, azaz a webkartográfia napjainkban egyre nagyobb hangsúlyt kap. Szinte naponta jelennek meg újabb térképes weboldalak, keresők, játékok, útvonaltervezők. Némelyikük statikus felépítésű, azaz csak annyiban térnek el egy egyszerű weboldaltól, hogy térképek képei is láthatók. Más oldalak dinamikusak, azaz a felhasználó interaktív módon befolyásolhatja a megjelenő információt. Ezen dinamikus oldalak egy része bármilyen web-böngészővel használható, míg más esetekben a használathoz valamilyen speciális beépülő programot (plug-in) kell először telepíteni.

A következőkben a beépülő modulok nem igénylő dinamikus webkartográfiai alkalmazások készítésének lehetőségeit tekintem át.

Egy weboldal dinamikussá tétele alapvetően kétféle módon érhető el: szerver vagy kliens oldali programokkal. Az első esetben a munkát a szerver (a weboldalt tároló számítógép) végzi, a felhasználó böngészője csak a végeredményt mutatja meg. A háttérben vagy egy térinformatikai program (pl. ArcIMS), vagy egyszerűbb esetben valamilyen szerver oldali script (PHP, ASP, Perl stb.) dolgozik. (A script olyan program, aminek közvetlenül a forráskódját értelmezi a feldolgozó számítógép, nincs lefordítva, ellentétben a bináris, azaz gépi kódra fordított programokkal.)

A kliens oldali programok a felhasználó számítógépén futnak, a böngésző felülete alatt. Ezek vagy scriptek (JavaScript) vagy bináris programok (pl. Java appletek).

A legtöbb esetben a szerver- és kliensoldali programokat együtt használjuk. Előbbiek végzik az adatok lekérését, feldolgozását, utóbbiak felelősek a megjelenítésért, és a felhasználóval való interakcióért.

Kiseb méretű és főleg non-profit alkalmazásoknál (pl. oktatási cél) fontos lehet az, hogy a felhasznált programok díjmentesen legyenek használhatók. Ilyen szempontból jó választás a

PHP/JavaScript kombináció, mellyel viszonylag egyszerűen lehet látványos eredményeket elérni.

A webkartográfiai alkalmazások vagy raszteres, vagy vektoros adatokkal dolgoznak. Raszteres adat lehet egy bedigitalizált vagy egyszerűen raszteres formátumban elmentett térkép, vagy esetleg légifotó, műholdkép. Ilyen adatok esetén a következő funkciókra lehet szükség:

- mivel a térkép az esetek többségében jóval nagyobb, mint a képernyő, a képen valahogy „navigálni” kell,
- nagyítás/kicsinyítés,
- tematikus rétegenként való megjelenítés,
- koordináta-meghatározás.

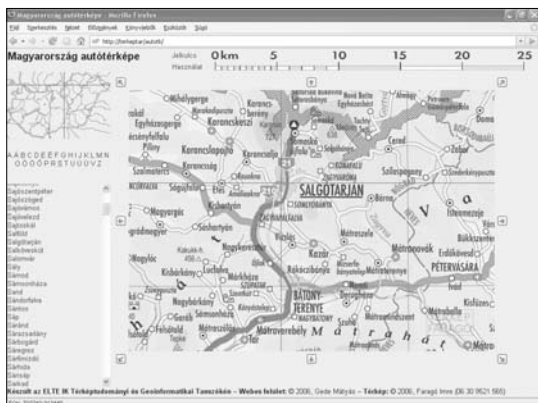
Vektoros adatokkal általában valamilyen gráfot írunk le, csomópontokkal és éllel. A pontok valamilyen helyet jelölnek, az él pedig egy lehetséges összeköttetést e pontok között. Bonyolultabb esetben az éllekhöz is tartozhat geometriai információ (a végpontjaik helyzetén kívül is): ha például egy út vagy egy vízfolyás egy szakaszát jelképezik, akkor egy pontsorozat jelöli a nyomvonalat. Ilyen adatok esetén alapvetően két feladat merül fel:

- az adatokból valamilyen térképszerű ábrázolás generálása,
- adott pontok közötti valamilyen szempontból optimális útvonal keresése.

## 2. Raszteres alkalmazások

### 2.1. Nagyméretű képek ábrázolása

Nagyméretű képek (térképek, légifotók stb.) webes megjelenítésénél valahogyan fel kell oldani a nagy képméret és a kis képernyőfelület közti ellentmondást. Mivel általában nem kívánatos az egész kép egy állományként hozzáférhetővé tétele, a kliens oldalra mindig csak a megfelelő képrészleteknek kell eljutniuk. Erre az egyik lehetőség, hogy a térképet azonos méretű kis téglalapokra bontjuk, ezeket külön eltároljuk, és mindig azt a néhányat rajzoltatjuk ki, ami éppen

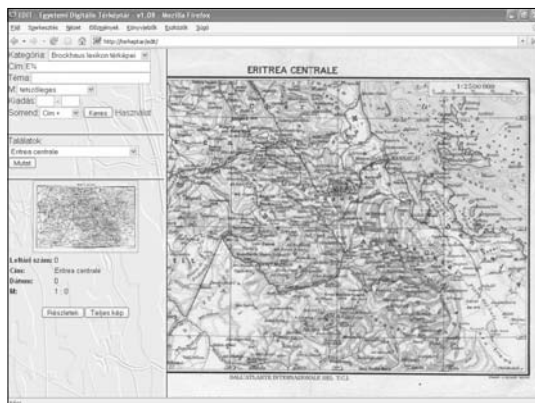


Magyarország autótérképe  
(<http://terkeptar.elte.hu/autotk/>)

a kívánt területre esik. Így működik például az általam készített „Magyarország autótérképe” című weboldal (<http://terkeptar.elte.hu/autotk/>). A térkép 100×100 pixel méretű darabokra lett vágva, és mindig az ablak méretének megfelelő mennyiségű (pl. 4×6 db) kép kerül ábrázolásra. A módszer előnye, hogy gyors, viszont nem készíthető tetszőleges méretű kivágat, csak a kis képek oldalhosszúságának egész számú többszöröse lehet a képméret. A térkép esetleges aktualizálása esetén pedig újra el kell készíteni a mozaikdarabokat.

Egy másik lehetőség, hogy a nagy képet egyben tároljuk, és a nagy állományból alkalmanként vágjuk ki a megjelenítendő részletet. Ez egyszerűen megoldható a PHP nyelvhez (is) használható GD grafikus függvénycsomag segítségével. A megoldás előnye, hogy bármekkora részlet készíthető, azonban nagy képeknél a kivágás nagy erőforrásigényű. Így ez a módszer akkor célszerű, ha pl. nagy mennyiségű képünk van, amit nem gazdaságos kicsire darabolva tárolni, viszont nem túl nagy a „felhasználói aktivitás” (nincs túlságosan leterhelve a kiszolgáló szerver). Ilyen például az ELTE Térképtudományi és Geoinformatikai Tanszékének Egyetemi Digitális Térképtára (EDIT), amely webes felületen keresztül is elérhető.

A térképen való navigálást mindenképpen megkönnyíti egy kis áttekintő térkép alkalmazása, melynek bármelyik részére kattintva a megfelelő részlet jelenik meg nagyban. Ha csak egy térképről van szó, akkor célszerű ezt a kis térképet külön elkészíteni, erősen generalizálva, hogy jól áttekinthető legyen (ilyen készült pl. az autótérképhez). Ha nagy mennyiségű térképünk



Az Egyetemi Digitális Térképtár webes böngészőfelülete

van, akkor ez nem lehetséges, ilyenkor automatikusan kell az áttekintőt generálni a nagy képből. Az EDIT rendszernél ez a kis kép akkor generálódik, mikor egy térképet először néznek meg.

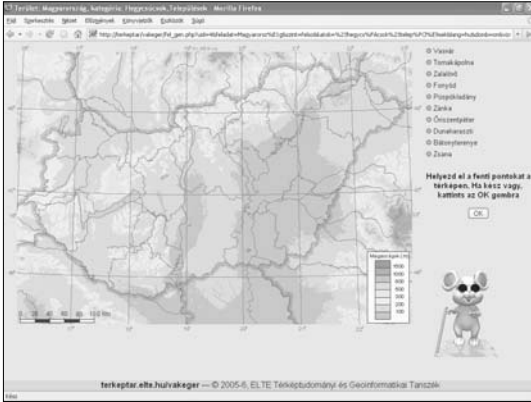
Az áttekintőtérképen való navigálást vagy szerver-oldali „image-map” segítségével (a képre kattintva az egérmutató koordinátái elküldődnek a feldolgozó scriptnek), vagy a képre vonatkozó egér-események JavaScriptben való feldolgozásával oldhatjuk meg. Célszerű a kiválasztott helyet meg is jelölni egy célkeresztrel vagy egy kis téglalappal. Ezek általában egy 1×1 pixel méretű kép megfelelő méretre nyújtásával/nagyításával, és az adott pozícióba helyezéssel oldhatók meg. A kijelölő téglalap esetén célszerű a PNG (Portable Network Graphics) formátum használata, amivel félig átlátszóvá tehető a kép.

## 2.2. Nagyítás, kicsinyítés

A nagyítás/kicsinyítés raszteres térképek esetén nehéz kérdés: ha kicsiben is élvezhető képet szeretnénk kapni, akkor több méretarányban kell elkészítenünk a térképet. Ettől függetlenül persze megoldható egy raszteres kép vagy képrészlet kicsinyítése: a már említett GD függvénykönyvtár megfelelő függvényei segítségével.

## 2.3. Térképi rétegek szelektív megjelenítése

Ha a különböző térképi rétegek külön-külön fájlban vannak tárolva (célszerűen GIF vagy PNG formátumban, mivel ezeknél definiálhatunk átlátszó szint), akkor ezekből tetszőleges kombinációjú térképet állíthatunk össze. Ehhez egyszerűen megfelelő sorrendben egymásra kell



A „Vakegér” játék  
(<http://terkeptar.elte.hu/vakeger>)

helyezni a képeket. Ilyen megoldással működik például a „Vakegér” nevű vaktérképés játékunk (<http://terkeptar.elte.hu/vakeger>), melynél a fókahálózat, hipszometrikus domborzat, vízrajz, megye- és országhatárok tetszőleges kombinációban választhatók.

#### 2.4. Koordináta-meghatározás

Ha a webes térképünkön valamilyen koordináta-meghatározást szeretnénk végezni, a raszteres képet előbb georeferálnunk kell. Ha ismerjük a térkép vetületét, és van néhány illesztőpontunk, akkor ez nem túl bonyolult feladat: az illesztőpontok képi koordinátái és vetületi koordinátáinak segítségével egy elsőfokú egyenletrendszer írható fel, mely megadja a leképezést a két koordináta-rendszer között. A vetületi koordinátákból pedig kiszámíthatjuk az alapfelületi koordinátákat (ha szükségünk van rájuk).

Ha a vetületet nem ismerjük vagy túl bonyolult, vagy esetleg más vetületi koordinátákra van szükségünk, akkor sincs probléma, ha elég sok illesztőpontunk van. A „Magyarország Autótérképe” esetében például EOv koordinátákat láthatunk. Ehhez illesztőpontokként a fókahálózati vonalak metszéspontjait használtam, melyek EOv és képi (pixel-) koordinátái is meghatározhatók. Egy adott ponton a hozzá legközelebb (általában köré) eső foktrapéz sarokpontjaihoz képest interpolálva számolódna a koordináták.

Ha a georeferálást megoldottuk, akkor azt kell eldönteni, hogy mikor és milyen adatokból határozzuk meg a koordinátákat. Lehetséges egyrészt a képrészletre kattintva, vagy az eget felette mozgatva a mutató alatti pixelkoordinátákból

vetületi vagy alapfelületi koordinátákat számítani és azokat a státuszsorban vagy egy felugró ablakban kiírni. Ehhez JavaScriptben kell lekezelni a megfelelő egér-eseményeket. Szükséges lehet a másik irányú konverzióra is: ha például egy pont koordinátáit ismerve annak helyét szeretnénk megmutatni a térképen. Ebben az esetben annyival bonyolultabb a helyzet, hogy először a kivágatot kell a megfelelő helyre állítani, és utána a pont helyét megjelölni.

### 3. Vektoros alkalmazások

#### 3.1. Adattárolás

Vektoros adatok esetén az első kérdés az adattárolás formája. Egyszerűbb esetben, mikor az adatkészlet viszonylag statikus (nincs rendszeres feltöltés, bővítés stb.), célszerű lehet valamilyen vektoros fájlformátum alkalmazása.

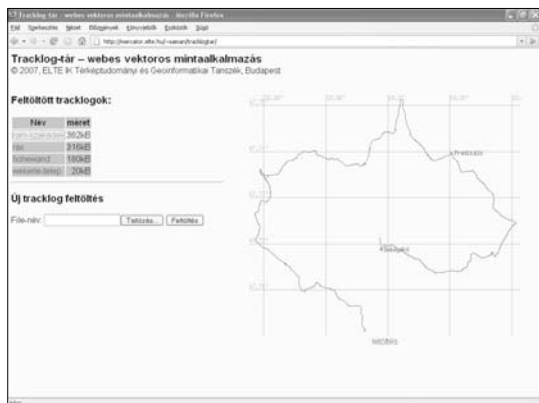
Ha azonban az adatkészlet gyorsan változhat (például a felhasználók is tölthetnek fel adatokat), vagy többféle szempont alapján is leválogathatóknak kell lenniük az adatoknak, akkor célszerű valamilyen vektoros adatformátumokat is támogató adatbázis-rendszer használata. A PHP-hez jól alkalmazható például a MySQL rendszer, ami ráadásul szintén szabadon felhasználható.

A MySQL ismeri a pont, a töröttvonal és a poligon adattípusokat, és számos hozzájuk kapcsolódó függvényt (pl. töröttvonal hossza, poligon területe stb.).

#### 3.2. Megjelenítés

Vektoros adatok grafikus megjelenítése a webben nem is olyan egyszerű feladat, mint elsőre gondolnánk, mivel a HTML formátum nem ismeri a vektoros rajzi elemeket (mint pl. vonalak, sokszögek stb.). Ezért két lehetőségünk van: vagy raszteres képet generálunk, melyet már beilleszthetünk a HTML oldalba, vagy valamilyen beágyazott objektumot használunk (Flash, Java). Ez utóbbiaknak az a hátránya, hogy használatukhoz a felhasználó böngészőjén telepíteni kell a megfelelő bedolgozó modulokat.

Raszteres képet viszont könnyen generálhatunk a vektoros adatokból PHP-ben a GD segítségével. A problémák ilyenkor ugyanazok, mint bármilyen esetben, amikor automatikusan generálunk adatokból térképet: a megjelenítendő elemek kiválasztása, az egymást fedő feliratok elkerülése stb.



Tracklog-tár: mintapélda a vektoros adatokból való térképgenerálásra PHP-ben

Mivel ezek a problémák csak jelentős programozási munkával oldhatók meg tisztességesen, a vektoros térképek megjelenítéséhez csak egyszerűbb esetekben célszerű script nyelveken alkalmazást fejleszteni. Egy ilyen egyszerű, de mégis használható példa az ábrán is bemutatott „tracklog-tár” alkalmazás, melyben a felhasználók feltölthetnek, letölthetnek GPS-es mérési adatokat, és letöltés előtt megnézhetik az adott tracklogból generált képet is.

### 3.3. Útkeresés

Ha van egy megfelelő úthálózat-gráfunk, akkor útvonal-tervező alkalmazás is készíthető, célszerűen térképi megjelenítéssel egybekötve. Az útvonal-tervezés viszonylag egyszerű addig, amíg a gráf mérete kicsi: szélességi keresést alkalmazva gyorsan megtaláljuk az optimális utat. Minél több attribútumot rendelünk a gráf éleire (melyek az egyes útszakaszokat reprezentálják), annál sokrétebb lehet a keresés. Például egy túraútvonal-tervezőnél az egyes útszakaszokhoz hozzárendelhető a menetidő egyik, illetve másik irányban, nehézségi fok, esetleg valamilyen speciális felszerelés szükségessége (lámpa, jégcsákány, kötél stb.).

## 4. Összefoglalás

Amint a bemutatott példák is látható, egyszerűbb feladatok esetén a különböző script nyelvek

alkalmazásával könnyen és gyorsan lehet jól működő webkartográfiai alkalmazásokat készíteni. Természetesen korlátai is vannak a módszernek: nagyon számításigényes feladatoknál a futás közbeni fordítás (interpretáció) miatt a scriptek lényegesen lassabbak lehetnek, mint a bináris programok. Előnyeik (ingyenes, szabad felhasználhatóság, gyorsan elérhető látványos eredmények) miatt azonban mindenképpen helyük van az internetes térképészetben.

## Creating Webcartography Applications using Scripting Languages

Gede, M.

### Summary

The possibilities and limits of the usage of scripting languages in the field of web-cartography are discussed in this paper. At first a short introduction is presented about the grouping of webcartography applications from different aspects, then the typical usages of scripts are shown with examples including browsable maps, a digital maproom database, and a mute map game.

### IRODALOM

1. *Guszlev Antal*: A webtérképezés környezete – lehetőségek és korlátok – Geodézia és Kartográfia, 2003. október
2. *Stegena Lajos*: Vetülettan – Tankönyvkiadó, Budapest, 1988.
3. *Szlávi Péter*: Előadás a gráftípusról – Mikrológia Szilánkok 5. – ELTE TTK Ált. Számítástudományi Tanszék kiadványa, Budapest, 1992.
4. Client-Side JavaScript Reference – <http://docs.sun.com/source/816-6408-10/contents.htm>
5. Cascading Style Sheets, level 2 – <http://www.w3.org/TR/1998/REC-CSS2-19980512/>
6. PHP kézikönyv – <http://www.php.net/manual/hu/>
7. GD Graphics Library – <http://www.boutell.com/gd/>
8. Mysql documentation – <http://dev.mysql.com/doc/>
9. HTML 4.01 Specification – <http://www.w3.org/TR/html4/>