

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
INFORMATIKAI KAR

A Tihanyi-félsziget interaktív webes térképe

DIPLOMAMUNKA
TÉRKÉPÉSZ MESTERSZAK

Készítette:

Fakan Péter Bálint

térképész mesterszakos hallgató

Témavezető:

Dr. Gede Mátyás

adjunktus

ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2016

Témabejelentő lap helye

TARTALOMJEGYZÉK

1. BEVEZETÉS	4
2. MEGLÉVŐ INTERNETES TÉRKÉPEK VIZSGÁLATA	5
2.1 OpenStreetMap	6
2.2 Turistautak.hu	7
2.3 OpenMaps.eu	8
2.4 Karancs webes turista-útikalauz	9
2.5 Mapy.hiking.sk	10
2.6 Mapa-turystyczna.pl.....	11
3. TERVEZÉS.....	13
4. ADATGYŰJTÉS- ÉS FELDOLGOZÁS	14
4.1 Adatgyűjtés.....	14
4.2 Adatfeldolgozás	16
5. KÖZZÉTÉTEL AZ INTERNETEN.....	18
5.1 A MapServer	18
5.2 A mapfile.....	19
5.3 Az OpenLayers API, mint a térkép megjelenítője.....	22
5.4 Interaktivitást biztosító funkciók	25
6. ÁBRÁZOLÁSI MEGOLDÁSOK- ÉS KORLÁTAIK.....	30
6.1 Turistautak ábrázolása.....	30
6.1.1 Ábrázolás kiemelt térképi színnel.....	32
6.1.2 Ábrázolás a jelzés színével azonos szín használatával.....	35
6.1.3 Ábrázolás kiemelt térképi színnel és jelentős útvonallal	37
6.2 A pontszerű objektumok ábrázolása	38
7. ÖSSZEGZÉS.....	39
8. FORRÁSJEGYZÉK	40
8.1 Irodalomjegyzék	40
8.2 Ábrajegyzék.....	42
9. KÖSZÖNETNYILVÁNÍTÁS.....	44
10. NYILATKOZAT.....	45

1. BEVEZETÉS

Diplomamunkám célja egy olyan nyílt forráskódú interaktív webes térkép elkészítése, amely a kirándulókat, utazókat segíti. Ez a munka tekinthető a korábbi – alapszakon készített – szakdolgozatom továbbfejlesztéseként, hiszen akkor egy papír alapú turistatérképet készítettem, most pedig ezt kívánom az interneten interaktív formában megjeleníteni. A munkám során ennek a papírtérképnek az adatait elsődleges információ-forrásként fogom felhasználni. A feldolgozás során a meglévő adatokra hagyatkozok, terepi felmérést nem fogok végezni.

Interaktív webes térkép kifejezés alatt egy olyan térképet értek, amely tartalmazza a turistautakat és tanösvényeket azok részletes adataival együtt, a természeti látnivalókat és az épített nevezetességeket. A térkép képes lesz megjeleníteni az épített nevezetességek információit, például az egyes múzeumok nyitva tartási idejét, vagy akár a belépőjegy árát is. Jelen esetben a Tihanyi-félszigetet tekinthetjük mintaterületnek, ugyanis a diplomamunkámban leírtak hasznosításával bármely területről készíthetünk hasonló térképet.

A diplomamunka térképe a nyílt forráskódú MapServer térinformatikai programmal készül el, az interneten pedig az OpenLayers megjelenítő segítségével teszem elérhetővé. A megjelenítés során figyelemmel vagyok arra is, hogy a felhasználó akár okostelefon böngészőjéből is elérheti a térképet, így erre a platformra is érdemes optimalizálni a weboldalt. Munkám során megvizsgálom a már meglévő internetes térképeket, ábrázolási módszereiket, interaktivitási lehetőségeiket, szemléltetem az egyes térképek pozitív és negatív megoldásait. Ismertetem az általam elkészített térkép alapanyagait, a térkép elkészítésének folyamatát, és megoldást próbálok találni az egyes ábrázolási korlátok áthidalására. Többféle megjelenésű térképeket készítek, és ezekből igyekszem kiválasztani a legkevesebb korláttal rendelkező megoldást, ami a végtermék lesz.

2. MEGLÉVŐ INTERNETES TÉRKÉPEK VIZSGÁLATA

A diplomamunka első szakaszában megvizsgálom az interneten jelenleg is megtalálható webes térképeket, ismertetem azok – általam megítélt – pozitív és negatív megjelenítési megoldásait. A vizsgálat során kitérek mind magyar, mind külföldi – szlovák és lengyel –, illetve nemzetközinek tekinthető internetes térképi ábrázolásokra. Összesen hat webes térképet fogok összehasonlítani a munkafolyamat során, ezekről az alábbi felsorolásban adok áttekintést:

- OpenStreetMap (OpenStreetMap, 2016)
- Turistautak.hu térképportál raszteres térképnézegető (Magyar Geocaching Közhasznú Egyesület 2002-2016)
- OpenMaps.eu raszteres kimenete (OpenMaps Alapítvány, 2007-2016)
- Karancs webes turista-útikalauz (Varga, 2014)
- Szlovák mapy.hiking.sk (Mapový Portál Hiking.sk, 2002-2016)
- Lengyel mapa-turystyczna.pl (Sygnatura, 2015)

2.1 OpenStreetMap

Az OpenStreetMap egy ingyenes világtérkép, amelynek Magyarországról is jól felépített adatbázisa és online térképe van. Ez a térkép azért ingyenes, mert a térképi információk közösségi felmérésen és munkán alapulnak, nem pedig egy cég saját terméke. Az ingyenes internetes világtérkép, „a szabad adathalmaz” tulajdonosa az OpenStreetMap Alapítvány, tehát a térkép felhasználásakor vagy módosításakor az általuk megszabott „Creative Commons Attribution-ShareAlike 2.0” (CC BY-SA) licenc szerint kell eljárni. A licenc értelmében az OpenStreetMap megosztható bármely formátumban, illetve átalakítható vagy az adatait felhasználva internetes térkép vagy bármely más szolgáltatás építhető belőle. Böngészhető elektronikus térképek esetében a térkép jobb alsó sarkában az „© OpenStreetMap közreműködők” hivatkozás kötelezően feltüntetendő, mely az általam készített térképekre is érvényes. Ez a térkép jó kiindulási alap lehet bármely online böngészhető térkép elkészítéséhez. Alapvetően interaktív felületet biztosít, de ez az interaktivitás csak a térkép felhasználói mozgásáig, illetve kicsinyítés- és nagyításig tart. A térkép bal felső sarkában egy keresődoboz található, mellyel az OpenStreetMap adatbázisában lévő rekordokra kereshetünk, találat esetén ránagyít a térkép a keresett objektumra. A térkép csak kótás domborzatábrázolást biztosít, tehát csak a hegycsúcsok és azok magassági adatai ábrázoltak. Szintvonalas domborzatábrázolás nem található rajta, domborzatárnyékolás pedig szintén nincsen, ami egy nagy szintkülönbséget bemutató térképen hasznos lehetne. A térkép tartalma részletes, gazdag út- és utcahálózat, közlekedési hálózat található rajta. Megfelelő nagyítási szinten rengeteg nevezetesség látható a térképen. Ez a részletesség a közösségi térképezésnek köszönhető, e nélkül jó eséllyel kevésbé lenne részletgazdag az OpenStreetMap. A térkép bal alsó sarkában egy dinamikusan változó mértékléc található, amely a nagyítási szintek váltakozásának megfelelően változik, azonban konkrét méretarányt számmal nem definiál. A weboldal elérési útvonala is dinamikusan változó, amely a webcímen kívül az aktuális nagyítási szint értékét, illetve a képernyő közepén lévő pont földrajzi koordinátáit tartalmazza. A diplomamunka témájául választott térképi alkalmazáshoz az OpenStreetMap önmagában kevésbé interaktív, azonban háttértérképként kiválóan felhasználható.

2.2 Turistautak.hu

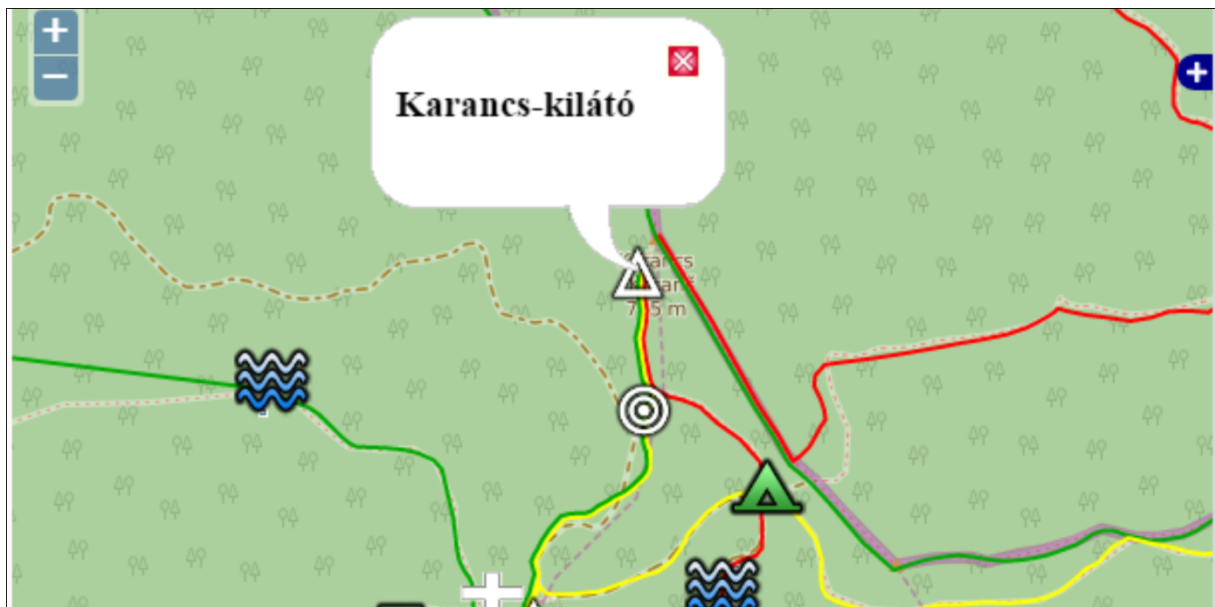
A Turistautak.hu térképportál elsősorban Magyarország turistaút-hálózatát hivatott teljes körűen felmérni, emellett a teljes magyarországi közúthálózatot, a települések utcaszintű felmérését is végzi. Ezen felül helyeket (POI) gyűjt és csoportosít, illetve ezeket szabadon hozzáférhető adatbázisában publikálja. A portál munkája szintén a nonprofit, közösségi térképezésen alapul. A weboldal tulajdonosa a Magyar Geocaching Közhasznú Egyesület, amely az oldal térképi adatbázisát az „Open Database License” (ODbL) alapján teszi nyilvánossá. A térképi adatbázis a feltöltött GPS nyomvonalak és a letölthető térképek összessége. A portál alapvetően elektronikus térképeket kínál letöltésre, amelyeket számítógépen keresztül a megfelelő célszoftverekkel (Garmin MapSource, Garmin BaseCamp) a GPS eszközökre tölthetünk. Ezen túlmenően kétféle webes raszteres térképi kimenettel is rendelkezik a weboldal. A régebbi, időben korábban megjelent verzió, egy a weboldalba beágyazott térkép, melynek kezelőszervei szintén az oldalba vannak beépítve. Ezzel a szolgáltatással változtathatjuk a térkép méretarányát, településeket kereshetünk az adatbázisban, az egyes útszakaszokról információt kérhetünk le, illetve többféle formátumban (SVG, PDF, PNG) eltárolhatjuk az aktuális állapotot. Sajnos azonban ez a beágyazott térkép kicsi, és nehezen kezelhető. Az új verzió egy teljes képernyős, okostelefonokra optimalizált térkép, amely nem az eredeti weboldalban van beágyazva. Ez a térkép az OpenLayers segítségével jelenik meg. Előnye, hogy sokkal inkább felhasználóbarát, mint a korábbi változat. A megjelenített térképen szintvonalas domborzatábrázolás található a megfelelő méretarányokban, amelynek adatai nyilvánosan hozzáférhető SRTM domborzatmodellből származnak. A domborzatárnyékolás nem alapértelmezett beállítás, de a rétegekapsolóban tetszés szerint használható. További pozitív tulajdonsága, hogy a térképen dinamikus mértékléc található, illetve konkrét számértékkel is kiírja az aktuális méretarányt. A térkép jobb alsó sarkában látható a kurzor által jelölt helyen a földrajzi szélesség és hosszúság. Alapvetően szerintem ez egy jó internetes megjelenítési megoldás, de a térképi objektumok nem kattinthatók, információt nem hordoz, véleményem szerint kevésbé informatív.

2.3 OpenMaps.eu

Az OpenMaps Projekt egy 2007-ben indult szerveződés, melynek célja, hogy ingyenes, digitális formátumú Közép-Európa térképeket tegyen elérhetővé. 2012 óta a weboldal és a kezdeményezés tulajdonosa az OpenMaps Alapítvány, így a térképek felhasználásakor, esetleges módosításakor az általuk előírt licencek alapján kell eljárni. A letölthető digitális térképekre az OpenStreetMapre is érvényes CC BY-SA 2.0 vonatkozik, minden más OpenMaps termékre (online raszteres térképnézegető, útvonaltervező) a „Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported” (CC BY-NC-SA 3.0) licenc az irányadó. A magyarországi térképi adatok felhasználásakor a Turistautak.hu szabályai az érvényesek, ugyanis a térkép a két társaság együttműködésével készült. A Magyarországon kívüli területeket lefedő térképekre az ODbL licenc vonatkozik. Az OpenMaps térképei Magyarország, Ausztria, Csehország, Szlovákia, Románia, Bulgária, Szerbia, Montenegró, Horvátország, Bosznia-Hercegovina, Szlovénia és Liechtenstein teljes államterületét lefedik, illetve kis részlettel rendelkeznek Lengyelország, Ukrajna, Olaszország, Svájc és Németország területéről. A raszteres térképnézegető egy OpenLayers segítségével megjelenített interaktív webes térkép. A felhasználó által a térkép böngészhető, kicsinyíthető-nagyítható a beépített nagyítási csúszka segítségével. A térképen a felhasználó útvonalat jelölhet ki kezdő- és végponttal, valamint köztes pontok beiktatásával is, a program pedig megtervezi a legrövidebb útvonalat, és jelöli azt a térképen. Az útvonal adatai, mint távolság, szakaszidő és szintkülönbség felugró ablakban jelennek meg. Tartalma a fentebb részletezett két térképéhez hasonló, megfelelő nagyítási szinten a nevezetességek is láthatók, sajnos ezek már nem interaktív formában. A térképi domborzatábrázolást az árnyékolás jelenti. A térkép mögött SRTM adatokból származó digitális domborzatmodell van beépítve, amely a szintkülönbségek kiszámítását és az árnyékolás elkészítését szolgálja. A térképen megtalálhatóak az aktuális földrajzi koordináták és a méretarány konkrét számértékkel jelezve. A méretarányt dinamikus mértékléc is szemlélteti. A rétegkapcsolóban a felhasználó több lehetőség közül választhat. Alapvetően ez a térkép már interaktív, de szintén kevésbé informatív a mögötte lévő adatbázist illetően.

2.4 Karancs webes turista-útikalauz

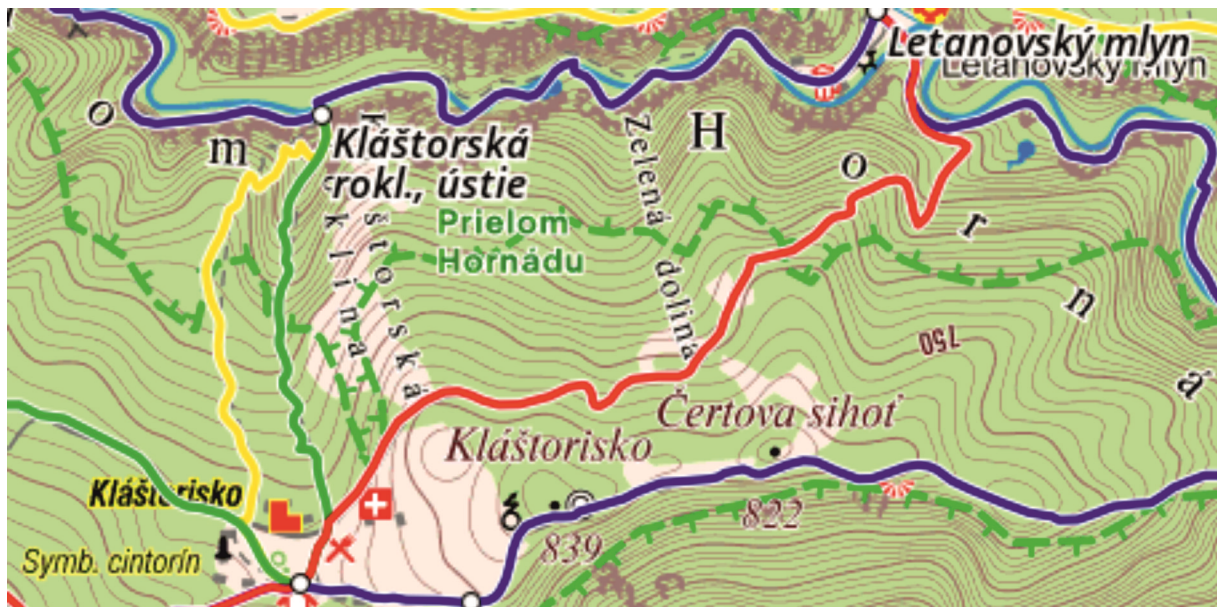
A Karancs webes turista-útikalauz Varga Balázs korábbi, 2014-es diplomamunka végterméke. Az útikalauz egy interaktív internetes térkép, amely a Karancs-hegység magyarországi és szlovákiai oldalán is információt ad a túraútvonalakról, illetve a terepen megtalálható érdekességekről. A térkép OpenLayers segítségével van megjelenítve, illetve beágyazva a weboldalba. Az oldal közepén lévő kis térkép a mobiltelefonok kijelzőjén kevésbé jól látszik, ezt mindenképp teljes oldalasként kellene megoldani. A térkép viszont interaktív, tehát felhasználó által böngészhető, mozgatható, valamint kicsinyíthető, nagyítható. A térképen a szimbólumokra kattintva a hozzájuk tartozó információ felugró ablakban jelenik meg (1. ábra), csakúgy, mint a kijelölt turistaútvonalak esetében, ahol az útvonal részletes információi olvashatók. A térkép tartalma véleményem szerint kellően részletes, a kirándulás előtt böngészve elég jó ismeretet lehet kapni a hegységről. A rétegkezelőben több ismert térképet tehetünk be háttértérképnek, mint például a Google Maps és az OpenStreetMap különböző típusait. A túraútvonalak és helyjelzők külön rétegen kaptak helyet, amelyek a felhasználó szerint tetszőlegesen ki-bekapcsolhatók. Domborzatábrázolás a térképen nincsen. A kurzor aktuális pozíciójának koordinátái a térképen nem láthatók, mértékléc, méretarány sincs feltüntetve. Ez a térképi megoldás már közel áll az általam kigondolt térképhez, kellően interaktív és információgazdag a felhasználó számára.



1. ábra: Karancs webes túra-útikalauz OpenStreetMap alapon

2.5 Mapy.hiking.sk

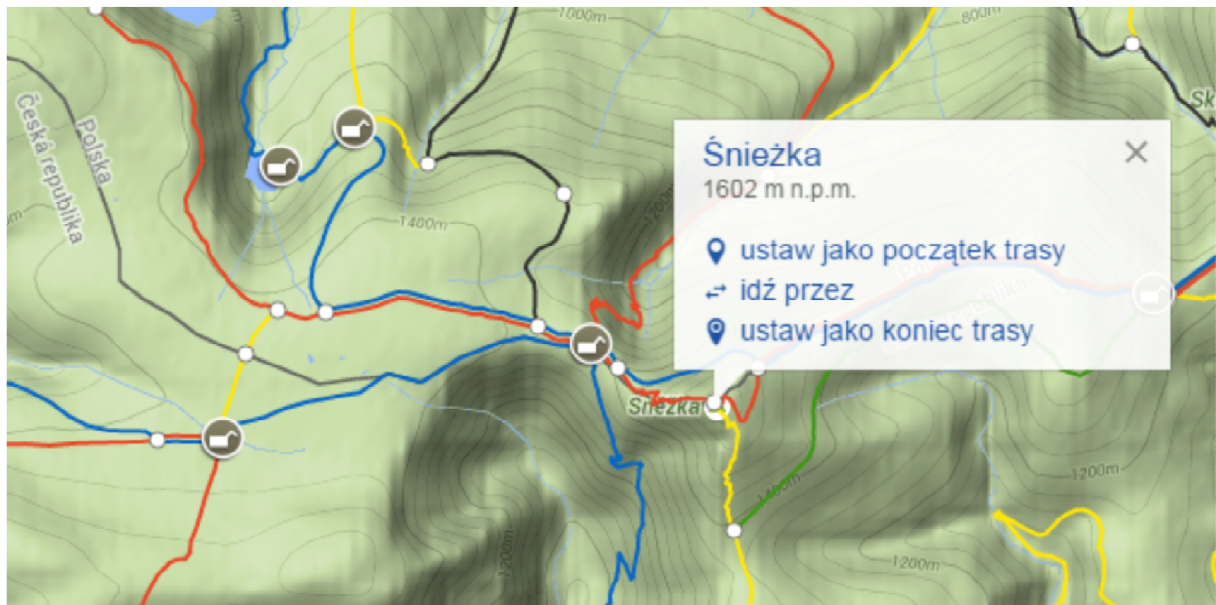
A Mapy.hiking.sk egy online raszteres turistatérkép, amelyet a szlovák Hiking.sk weboldal fejlesztői építettek fel. A térkép OpenLayers segítségével jelenik meg a weboldalon, alaphelyzeti háttértérképe a Google Terrain, azaz a Google domborzati térképe. Ezen felül választható a Google Műholdtérkép és az OpenCycleMap domborzati térképe is. Nagy méretarányban, vagyis az utolsó három nagyítási szinten a cseh térképkészítő cég, a SHOCart turistatérképét böngészhetjük (2. ábra). A rétegkapcsolóban sokféle opció közül választhat a felhasználó. Az oldalon nemcsak a turistatérkép jeleníthető meg, hanem a Hiking.sk és a Szlovák Hegyi Mentőszolgálat (HZS) közös munkájaként létrejött lavinatérkép is. Ez a térkép is alapvetően interaktív, változtathatóak a rétegek vagy akár nagyíthat a felhasználó tetszés szerint. A pontszerű objektumokra kattintva információ ugrik elő a jobb oldali dobozban, azonban a vonalas objektumok nem interaktívak, nem látható semmilyen adat. Az oldal jobb oldalán a szlovákiai hegységeket tartalmazó lista található, amelyből egyet kiválasztva, az adott hegység körvonala kirajzolódik a térképen. A térképhez tartozik egy útvonaltervező funkció is. A térkép bal alsó sarkában a nagyítás függvényében változó mértékléc kapott helyet. Ez a megoldás is tartalmaz interaktív részeket, bár véleményem szerint egy online turistatérkép a turistautakról is szolgálhatna információval, ez a térkép pedig csak megjeleníti azokat, de részletes leírást nem ad róluk.



2. ábra: SHOCart turistatérkép a mapy.hiking.sk weboldalon

2.6 Mapa-turystyczna.pl

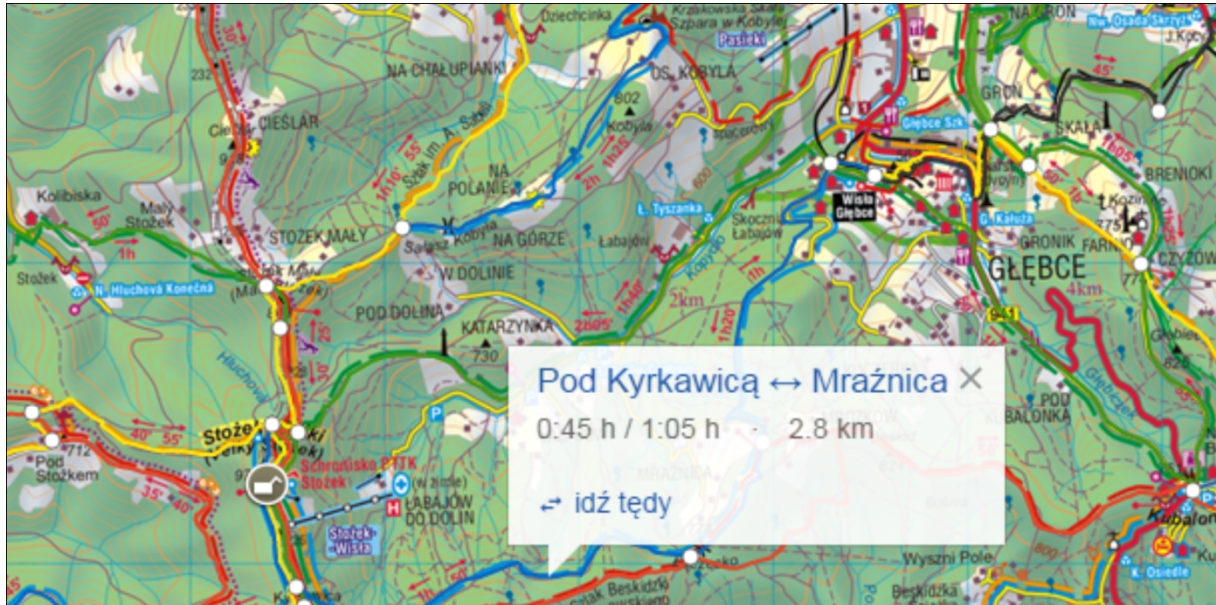
A mapa-turystyczna.pl weboldal egy lengyel kezdeményezés, amely elsősorban a lengyelországi turistautakat, és a túrázókat segítő pontokat hivatott felmérni. Nemcsak Lengyelország területéről rendelkezik térképpel, hanem a lengyel államhatárokon húzódó hegységek esetében a szomszédos állam területén haladó túraútvonalakról is jó képet ad (Magas-Tátra, Szlovákia / Óriás-hegység, Szudéták, Csehország). Az online turistatérkép megjelenítéséről a Google Maps API (Application Programming Interface) gondoskodik. Alaphelyzetben a háttértérkép a Google Terrain, azaz a Google domborzati térképe domborzatárnyékolással (3. ábra). A háttértérkép átállítható Google Műhold illetve a lengyel Sygnatura térképkészítő cég turistatérképére.



3. ábra: Interaktív térkép Google Terrain háttérrel és felugró ablakkal

A Sygnatura térkép csak nagy méretarány esetében böngészhető. Kis méretarányban térképi tartalomként csak a tájegységek nevei láthatóak, egy nagyítási szinttel feljebb pedig a kistájegységek neveit láthatjuk. Megfelelően nagy méretarány elérésekor megjelennek a turistautakat jelző színes vonalak, ahol a vonal színe megegyezik a terepen megtalálható jelzés színével. Még nagyobb méretarányban megjelennek a turistautak csomópontjaiban található interaktív jelölők, valamint a menedékházak és az egyéb fontos turistákat tájékoztató tereptárgyak. Ezek a terepen, az útelágazásokban található információs táblákat jelölik, illetve annak tartalmát teszik online elérhetővé. Nagy méretarányban a Google Terrain szintvonalas domborzatábrázolással is el van látva, csakúgy, mint a Sygnatura térkép is. A térképi tartalom teljesen interaktív, mind a vonalas objektumok, mind a pontszerű jelek információt hordoznak, ezek kattintáskor felugró ablakban jelennek meg. A turistautakról megtudhatjuk, hogy az adott sza-

kasz honnan hova tart, mekkora a távolság és mennyi idő szükséges az út megtételéhez (4. ábra). A pontszerű objektumokról megtudhatjuk a tereptárgy nevét és tengerszint feletti magasságát. A térkép mellett bal oldalon egy útvonaltervező- és objektumkereső panel található, amely a felhasználó igénye szerint kinyitható és becsukható. A weboldalnak mobiltelefonra optimalizált változata is van, illetve az Android alkalmazással akár élőben láthatjuk a pozícióinkat az okostelefonokon. Véleményem szerint ez egy kiváló interaktív térkép, a saját térképemet is ehhez hasonlóan fogom megalkotni.



4. ábra: Interaktív térkép Sygnatura turistatérkép háttérrel

3. TERVEZÉS

Mielőtt belevágnék az adatgyűjtésbe és a feldolgozásba, valamint a térkép elkészítésébe, az első feladat a leendő térkép megtervezése. Célom egy olyan webes térkép és a hozzá tartozó weboldal publikálása, amely kellően interaktív ahhoz, hogy a felhasználó könnyedén tudja kezelni és a számára szükséges információt kinyerni. A weboldalt a mai igényeknek megfelelően kell megterveznem, tehát úgy, hogy akár az okostelefonokon futó böngészőkben való megnyitáskor is jól nézzen ki, használható legyen egyszerűen.

A fentebb vizsgált minták közül a saját térképem leginkább a lengyel online turistatérképre fog hasonlítani. Az összes megvizsgált példa közül ezt találtam a legjobban kivitelezett, leginkább interaktív megoldásnak, ezért kívánom ezt alapul venni. Alapvető eltérés lesz, hogy míg a lengyel verzió a Google Maps API segítségével készült, addig én az OpenLayers API függvénykönyvtárat fogom használni. A térképi tartalom megjelenítéséhez és feldolgozásához a MapServer térinformatikai programcsomagot fogom igénybe venni. A térkép készítésekor a mesterképzés során a „Nyílt forráskódú webes GIS” és az „Önálló geoinformatikai projektfeladat” kurzusokról elsajátított ismereteimet fogom felhasználni.

A háttérnek választhatóan az OpenStreetMap, az OpenCycleMap és az OpenTopoMap térképeket fogom segítségül hívni. Ezeket a felhasználó a rétegkezelő menüben tetszés szerint megváltoztathatja. Mivel elsősorban a korábbi szakdolgozatom turistatérképét kívánom online módon megjeleníteni, ezért alapvetően a turistatérképi tematikára koncentrálok. A térképre szintvonalas domborzatábrázolást fogok feltenni, ugyanis a háttértérképeken ez nem szerepel, az OpenStreetMap-en is csak kótás domborzatábrázolás látható. Ábrázolom a turistautakat, azokat interaktív információval – szakaszhossz, szakaszidő, turistajelzés – vétezem fel. A turistautak elágazásait is interaktív megoldással kívánom ábrázolni, melyeket szintén információval töltök fel. A turistatérképi tematika része az épített és természeti nevezetességek (kiegészítő információk), ezek is felkerülnek a térképre, amelyeket a felhasználó tetszés szerint ki-be kapcsolhat a rétegkezelő menüben, illetve lekérdezheti az adatbázisból a ponthoz tartozó információt. A térkép bal oldalán egy ki-be csukható panel lesz, melyben egy keresődoboz kap helyet, amely segítségével kereshetőek lesznek az adatbázis elemei. A panel továbbá a jelmagyarázatot és a térkép használati utasításait fogja még tartalmazni. A térkép ki fogja írni a kurzor aktuális pozíciójának földrajzi koordinátáit, illetve egy dinamikus mértékléceket, valamint számértékkel kijelzett méretarányt fog megjeleníteni.

4. ADATGYŰJTÉS- ÉS FELDOLGOZÁS

4.1 Adatgyűjtés

A térképi tematika elkészítése, illetve megjelenítése előtt szükség van az adatok felmérésére. A dolgozat elején kitértem arra, hogy terepi munkát a térkép elkészítésekor nem végzek, ezért a korábbi szakdolgozat térképét és annak adatbázisát használok fel elsődleges adatforrásként. A 2014-ben elkészült turistatérkép tematikájának forrása az általam kézi GPS eszközzel felmért tracklogok, és útvonalpontok. A mostani webes térkép tematikáját ezek az útvonalpontok fogják képezni, kiegészítve a turistautak elágazásainak pontjaival. Az adatokat a korábbi munka közben GDB (Garmin GPS Database), illetve az esetleges későbbi feladatok megkönnyítése érdekében GPX (GPS Exchange Format) formátumban mentettem el.

A jelen dolgozat során készített webes térkép alapja az ESRI (Environmental Systems Research Institute) által kifejlesztett és karbantartott shapefile. A shapefile-ok létrehozása és szerkesztése térinformatikai (GIS) szoftverekkel lehetséges. Ilyen elsősorban szintén az ESRI által kifejlesztett ArcGIS térinformatikai programcsomag, vagy a nyílt forráskódú (Open Source) Quantum GIS. Mivel munkám során csak nyílt forráskódú programokkal oldom meg a térkép elkészítését és publikálását, ezért kézenfekvő és egyértelmű a Quantum GIS használata. Szükséges tehát a GPX fájlban tárolt adatokat SHP fájl formátumba konvertálni. A Global Mapper 13 szoftver képes a megnyitott GPX állományokat egyenesen ESRI shapefile-ba exportálni. Másrészt a Quantum GIS natívan nyitja meg és jeleníti meg a GPX állományokat. A programban erre a „Vektor > GPS > GPS eszközök” menüben nyílik lehetőség, valamint akár az állomány egyéb konverzióit is elvégezhetjük itt. Így az adatokat könnyedén menthetjük shapefile formátumba a megfelelő vetület kiválasztása után.

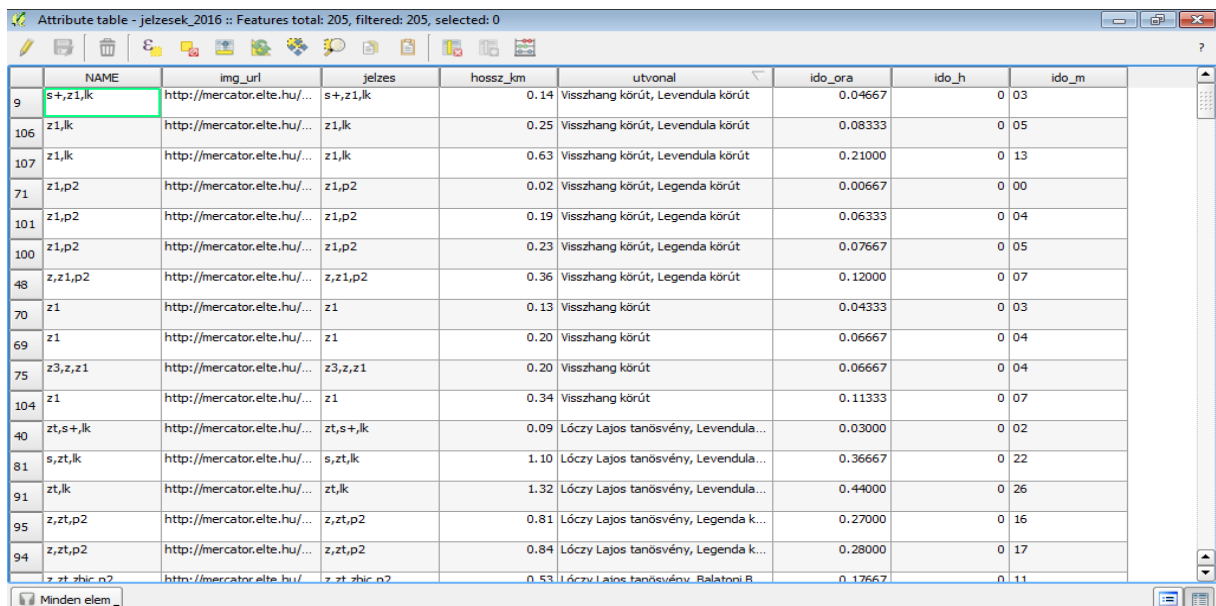
A vonalas objektumok adataihoz kétféleképpen juthattam hozzá. Egyrészt a <http://geofabrik.de> szerverről letöltöttem a Magyarországra szűkített OSM kivonatot ESRI shapefile formátumban. Ebből ki tudtam nyerni az összes olyan utat, amelyen az OpenStreetMap adatbázisa szerint nincs turistaút, a valóságban pedig van. Másrészt a <http://overpass-turbo.eu/> weboldalról letöltöttem a magyarországi turistautakat az OpenStreetMap natív, nyers fájlformátumában, amely az OSM XML. Az Overpass Turbo egy webes alapú adatszűrő szolgáltatás, mely az OpenStreetMap adatait dolgozza fel. Használatával speciális Overpass QL kóddal írott kéréseket küldhetünk az Overpass API-nak, amely az ennek pontosan megfelelő eredményt küldi vissza a felhasználónak (OpenStreetMap Dokumentáció, 2016; Overpass API Dokumentáció, 2016). A kapott adat nyers OSM XML állományként érkezik, amelyet át kell konvertálnunk shapefile-ba, hogy dolgozni tudjunk vele.

Az átalakítást az XMLparseOSM JavaScript programmal hajthatjuk végre, amely két *.kml* formátumú fájlt készít. Ezek a fájlok tartalmazzák a turistautakat vonalas elemekként. Meg kell adnunk a *.osm* kiterjesztésű fájl elérési útvonalát, illetve a két kimeneti fájl nevét a program indításakor. Az egyik KML fájlban minden útszakasz annyiszor szerepel, ahány turistajelzés van rajta, a másik fájlban pedig minden szakasz egyszer szerepel, de ekkor a turistajelzések attribútumként tárolódnak vesszővel elválasztva. A KML fájlokat a Global Mapper térinformatikai szoftverrel tudjuk megnyitni és egyből SHP formátumba exportálni. Az XMLparseOSM JavaScript program a GNU General Public License feltételeinek betartása mellett használható fel. A program az internetről szabadon letölthető az alábbi linken. (Gede, 2016)

https://github.com/samanbey/OSM_hiking_routes

4.2 Adatfeldolgozás

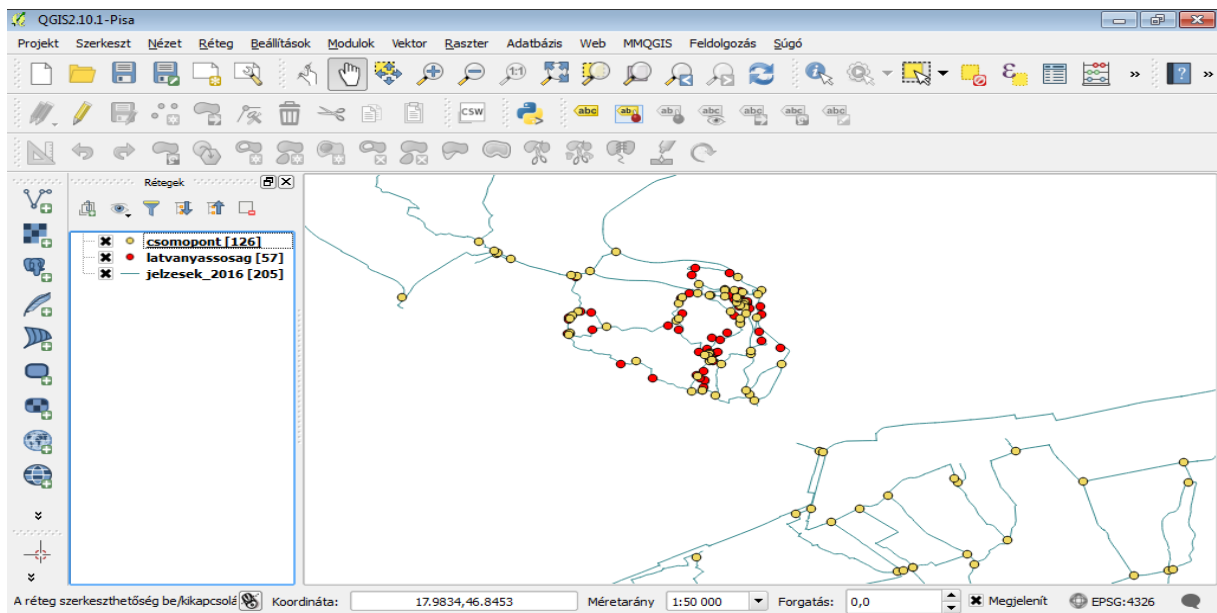
Az adatok összegyűjtése után azok rendszerezésére és feldolgozására került sor. A diplomadolgozathoz több különböző térképi megjelenítést készítettem, ezért többféleképp írtam meg az attribútumokat. Első esetben ott, ahol több turistaút fut közös nyomvonalon, a shapefile több egyedülálló vonalas objektumot tartalmaz ugyanazokkal a koordinátákkal (*friss_jelzesek.shp*). A *[name]* attribútum tartalmazza a turistautak jelzéseit, az *[img_url]* mező pedig az ezekhez tartozó PNG formátumú szimbólumot (5. ábra). A vonalas objektumok másik esete, amikor a közös nyomvonalon futó turistautakat egy vonalként tárolja a shapefile (*jelzesek_2016.shp*). Ekkor a *[name]* mező vesszővel elválasztva tartalmazza a turistautak jelzéseit (például p,s,z). Ebben az esetben az *[img_url]* mező egy URL címet tartalmaz, ahol egy PHP script generálja le a turistajelzések szimbólumait annak függvényében, hogy mi szerepel a *[name]* mezőben. Három vagy annál több jelzés esetén a jelek két oszlopba kerülnek. Amennyiben páratlan számú jelet hívunk meg, akkor az utolsó jelzés a többi alatt középre igazítva jelenik meg (Gede, 2016). A shapefile tartalmazza továbbá az útszakaszok hosszát kilométerben, az azok megtételéhez szükséges körülbelüli időadatot óra:perc formátumban és a túraútvonalak esetleges neveit (például Levendula körút).



	NAME	img_url	jelzes	hossz_km	utvonal	ido_ora	ido_h	ido_m
9	s+,z1,k	http://mercator.elte.hu/...	s+,z1,k	0.14	Visszhang körút, Levendula körút	0.04667	0	03
106	z1,k	http://mercator.elte.hu/...	z1,k	0.25	Visszhang körút, Levendula körút	0.08333	0	05
107	z1,k	http://mercator.elte.hu/...	z1,k	0.63	Visszhang körút, Levendula körút	0.21000	0	13
71	z1,p2	http://mercator.elte.hu/...	z1,p2	0.02	Visszhang körút, Legenda körút	0.00667	0	00
101	z1,p2	http://mercator.elte.hu/...	z1,p2	0.19	Visszhang körút, Legenda körút	0.06333	0	04
100	z1,p2	http://mercator.elte.hu/...	z1,p2	0.23	Visszhang körút, Legenda körút	0.07667	0	05
48	z,z1,p2	http://mercator.elte.hu/...	z,z1,p2	0.36	Visszhang körút, Legenda körút	0.12000	0	07
70	z1	http://mercator.elte.hu/...	z1	0.13	Visszhang körút	0.04333	0	03
69	z1	http://mercator.elte.hu/...	z1	0.20	Visszhang körút	0.06667	0	04
75	z3,z,z1	http://mercator.elte.hu/...	z3,z,z1	0.20	Visszhang körút	0.06667	0	04
104	z1	http://mercator.elte.hu/...	z1	0.34	Visszhang körút	0.11333	0	07
40	zt,s+,jk	http://mercator.elte.hu/...	zt,s+,jk	0.09	Lóczy Lajos tanósvény, Levendula...	0.03000	0	02
81	s,zt,k	http://mercator.elte.hu/...	s,zt,k	1.10	Lóczy Lajos tanósvény, Levendula...	0.36667	0	22
91	zt,k	http://mercator.elte.hu/...	zt,k	1.32	Lóczy Lajos tanósvény, Levendula...	0.44000	0	26
95	z,zt,p2	http://mercator.elte.hu/...	z,zt,p2	0.81	Lóczy Lajos tanósvény, Legenda k...	0.27000	0	16
94	z,zt,p2	http://mercator.elte.hu/...	z,zt,p2	0.84	Lóczy Lajos tanósvény, Legenda k...	0.28000	0	17
	z,zt,zhrc,p2	http://mercator.elte.hu/...	z,zt,zhrc,p2	0.53	Lóczy Lajos tanósvény, Balatoni B...	0.17667	0	11

5. ábra: Turistautak attribútum táblája a Quantum GIS programban

A pontszerű objektumokat tartalmazó shapefile-okban a turistautak csomópontjai (*csomopont.shp*), illetve a kategorizált látványosságok (*latvanyosság.shp*) találhatóak. A *csomopont.shp* tartalma az elágazás neve [*nev*], a tájegység [*tajegyseg*], és a tengerszint feletti magasság [*magassag*]. A *latvanyosság.shp* fájlban a pontszerű objektumok a [*tipus*] mező alapján kategorizáltak. Ennek az értéke lehet *kilato*, *muemlek*, *muzeum*, *tanallomas*, és *termeszet*i. Emellett tartalmazza az objektum nevét [*nev*], címét [*irsz*, *telepules*, *cim*], webes elérhetőségét [*web*], a belépőjegy árát, ahol elérhető [*belepo_f*, *belepo_gy*] és a nyitva tartást [*nyitva_tartas*]. A leendő webes interaktív térkép ezekben a fájlokban tárolt adatok alapján fog megjelenni (6. ábra).



6. ábra: A kész adatbázis Quantum GIS-ben

5. KÖZZÉTÉTEL AZ INTERNETEN

5.1 A MapServer

A MapServer egy nyílt forráskódú térinformatikai/kartográfiai programcsomag, amelyet eredetileg a University of Minnesota fejlesztett ki a NASA-val (National Aeronautics and Space Administration) közös együttműködésben. A MapServer jelenleg az OSGeo (Open Source Geospatial Foundation) tulajdona, melyet a világ minden táján élő programozók fejlesztenek és tartanak karban. A MapServer segítségével térképeket állíthatunk elő különböző formátumú vektoros és raszteres geoadatokból, interaktív formában publikálhatjuk az interneten. A legegyszerűbb esetben a MapServer működhet CGI (Common Gateway Interface) programként, de MapScript segítségével különböző szerveroldali szkriptnyelvek kiegészítéseként is funkcionálhat. Az egyszerű MapServer alkalmazás a következő elemekből épül fel:

- A mapfile a MapServer legfőbb eleme, amely definiálja, hogy a különböző adatforrások hogyan legyenek ábrázolva a térképen. Egyszerű szövegfájl, a kiterjesztése *.map*.
- Szükség van geoadatokra, amelyek lehetnek vektoros vagy raszteres állományok. Az alapértelmezett fájlformátum az ESRI shapefile. A MapServer azonban rengetegféle állományt támogat, ezek teljes listája a MapServer Dokumentációban érhető el.
- A HTML weboldal a kezelőfelület a felhasználó és a MapServer között.
- A MapServer CGI egy bináris vagy futtatható fájl, amely kéréseket fogad, és képet, térképet, adatot küld vissza válaszként. Ez a fájl a webservertől cgi-bin vagy scripts könyvtárban helyezkedik el.
- A webservertől vagy HTTP szerver szolgálja ki a felhasználó által megnyitott – a szerveren tárolt - HTML weboldalakat. (Gede, 2016; MapServer Dokumentáció, 2016)

5.2 A mapfile

A mapfile a MapServer lelke, amely leírja, hogyan ábrázoljuk az adatforrást, hol keresse a program az adatokat, illetve definiálja az objektumok közötti kapcsolatot. A mapfile egymásba ágyazott objektumokból áll, melyek mindegyike az objektum típusát jelző kulcsszóval kezdődik, és az END kulcsszóval végződik (Gede, 2016). A mapfile mindig tartalmaz egy MAP objektumot, amely leírja a térkép általános tulajdonságait, és meghatározza a térképi rétegeket. Hierarchikus felépítésű, az első mindig a MAP objektum, minden más csak ezután következhet. A MapServer a mapfile értelmezését a fájl elejétől a vége felé tartó sorrendben végzi, amely a gyakorlatban azt jelenti, hogy a korábban definiált térképi rétegeket a MapServer előbb rajzolja ki, mint az utána meghatározottakat. A MapServer 5-nél korábbi verzióiban maximum 200 térképi réteg definiálható egy mapfile-on belül, ennél újabb kiadások esetében nincs felső korlát. Az adatforrás elérhetősége lehet abszolút illetve relatív is. A mapfile nem kis- és nagybetű érzékeny, tehát elvben mindkét megoldás alkalmazható. A # jel mögötti tartalmat a MapServer nem használja, itt a könnyebb megértés érdekében szöveges megjegyzések írhatók. (MapServer Dokumentáció, 2016)

A mapfile rétegekből és osztályokból, valamint a térkép megjelenítési beállításáiból épül fel. A mapfile elején meg kell adnunk az alapbeállításokat, amely tartalmazza a kezdőelemet (#), a mapfile nevét, a rétegek státuszát, a képméretet, az image típusát, a térkép határait, a betűtípus-készletet, a térkép vetületét és a WMS (Web Map Service) működéséhez szükséges beállításokat. A térkép vetületét a PROJECTION objektumban kell megadni. Ezt követően egy átlátszó háttér van definiálva, majd ezután a szimbólumok meghatározása következik. Az alábbi kódrészlet ezt mutatja be:

```
#
MAP
  NAME "Jelzett_utak_csf"
  STATUS ON
  SIZE 800 600 # képméret pixelben
  IMAGETYPE png
  EXTENT 17.75 46.75 18.05 47.05 # a térkép határai
  FONTSET "libfonts.list.txt"
  PROJECTION
    "init=epsg:4326" # WGS-84
  END
  WEB
  METADATA
    "wms_enable_request" "*"
    "wms_feature_info_mime_type" "text/plain"
    "wms_feature_info_mime_type" "text/html"
```

END
END

A mapfile-on belül a LAYER objektummal definiáljuk a térképi rétegeket. Létrehozhatunk vektoros és raszteres térképi réteget is. Vektoros réteg esetében a típus TYPE lehet *point*, *line*, *polygon* és *annotation*. Raszteres réteg esetében a típus TYPE értéke *raster*. Mindkét esetben a DATA használatával adhatjuk meg a MapServernek az adatforrás elérhetőségét. Ha nem shapefile a forrásfájlunk, akkor a CONNECTIONTYPE beállításával egyéb – a MapServer által támogatott – vektoros állományokat használhatunk natív formátumukban. A különböző vektoros állományokat az OGR nyílt forráskódú könyvtár tartalmazza, ezért erre hivatkoznunk kell. Az OGR a GDAL (Geospatial Data Abstraction Library) könyvtár része, amelyet jelenleg szintén az OSGeo felügyel. Az alábbi kódrészlettel a MapInfo *.tab* állományban tárolt adatokat tudjuk a MapServer által olvashatóvá tenni:

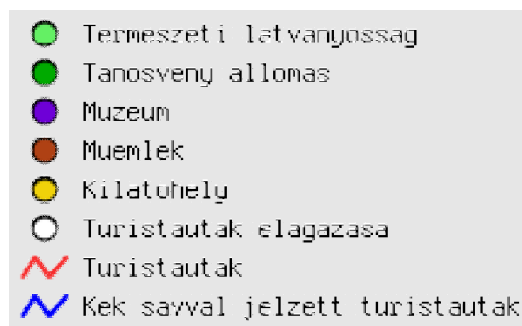
```
CONNECTIONTYPE OGR  
CONNECTION "autopalya.tab"
```

Amennyiben ezt nem állítjuk be, a MapServer alapértelmezésben az ESRI shapefile formátummal fog dolgozni. Az *annotation* olyan típusú réteg, amely a pontszerű elemeit csak akkor rendereli, ha azok elférnek a térképen az adott méretarányban. A LAYER objektumon belül osztályokat hozhatunk létre, ezeknek a stílusinformációit a CLASS és a STYLE elemek tartalmazzák. Az osztályozásnak akkor van nagy haszna, ha az adatforrás nem csak geometriai információt tartalmaz, hanem adatmezőket is. A különböző osztályok elemeit többféle módon jeleníthetjük meg (Gede, 2016). A CLASS objektumon belül az EXPRESSION kifejezéssel definiálható, hogy az elemek milyen feltételnek megfelelően kerüljenek az adott osztályba. A MapServer három féle EXPRESSION típust ismer fel, ezek a string összehasonlítás, a reguláris és a logikai kifejezés. A logikai kifejezések használata a legtöbb esetben kerülendő, ugyanis az ennek megfelelő térkép sokkal lassabban rajzolódik ki. Felhasználhatók a forrásban lévő adatmezők is, ekkor a mező nevét szögletes zárójelben kell feltüntetni. A CLASSITEM segítségével megnevezhetjük a konkrét attribútumot, melyet az EXPRESSION használatával vizsgálunk. Az osztályokat méretarányfüggően is meg tudjuk jeleníttetni a MapServerrel. A MINSCALEDENOM illetve MAXCALEDENOM kifejezésekkel definiálhatjuk, hogy az adott osztály elemei milyen méretarány-korlátok között jelenjenek meg (Gede, 2016).

A vektoros rétegeket a fentebb ismertetett TYPE meghatározásával hozhatjuk létre. A réteg tartalmát képező elemek stílusát a CLASS objektumon belül a STYLE segítségével definiálhatjuk. A STYLE használatával beállíthatjuk az elemek méretét, a vonalak szélességét, szaggatottságát, színét, háttérszínét, a körvonalak színeit, eltolhatjuk a vonalakat egymás mellett párhuzamosan, de itt határozhatjuk meg a szimbólumokat is a SYMBOL objektummal. A

szimbólumokat a megfelelő paraméterezéssel kézzel is felépíthetjük, de külső forrásból származó piktogramokat, képszerű jeleket is importálhatunk. A szimbólumot a mapfile-on kívül is definiálhatjuk, ekkor a külső fájlt a SYMBOLSET segítségével köthetjük a mapfile-hoz. Több térkép készítésekor ugyanannak a jelkulcsnak a használata esetében javasolt ez a megoldás. A kézzel készített szimbólum típusait a TYPE tartalmazza, mely lehet *ellipse*, *hatch*, *pixmap*, *svg*, *truetype* vagy *vector*. A külső forrásból származó raszteres képfájl (*pixmap*) az URL cím megadásával csatolhatjuk a szimbólumhoz.

A saját forrásból (például adatmezőből) származó térképi névrajzot képes megjeleníteni a MapServer. A névrajz beállításait szintén a mapfile-on belül kell elvégeznünk. A LABEL a réteg objektumon belül erre szolgál. A LABELITEM segítségével meghatározható, hogy mely attribútum alapján feliratozzuk a térképet, az osztályon belül pedig a LABEL paraméterezésével állíthatjuk be a betűtípusokat, színeket és méreteket. A LABEL használatával nem csak szöveges formátumú névrajz írható ki a térképre, de adatmezőben megadott URL címen elérhető raszteres kép is megjeleníthető. Ekkor az adatmezőre szögletes zárójelben kell hivatkoznunk. A névrajzi elemeket az OFFSET segítségével tolhatjuk el a hozzá tartozó elemtől. A POSITION megfelelő beállításával megadhatjuk, hogy a névrajz a hozzátartozó elemhez képest hol helyezkedjen el. A névrajz kiírása mellett a térképhez jelmagyarázatot is generáltathatunk a MapServerrel, amennyiben a CGI programot a *&mode=legend* paraméterrel hivatkozzuk meg (7. ábra). A jelmagyarázatban minden olyan osztály szerepel, ahol a NAME objektumot feltöltöttük névvel. A CGI program egy PNG formátumú képet készít, amelyet beszúrhatunk a megfelelő HTML kóddal a térképünk mellé. (Gede, 2016; MapServer Mapfile Dokumentáció, 2016)



7. ábra: MapServer által készített jelmagyarázat

5.3 Az OpenLayers API, mint a térkép megjelenítője

Az OpenLayers egy felhasználó oldali JavaScript függvénykönyvtár, amely használatával térképi adatokat jeleníthetünk meg a legtöbb mai böngészőben szerver oldali függőség nélkül. A keretrendszernek nem része térképi adat, csak a megjelenítésre szolgál. Az adatforrásunk rendkívül sokféle lehet, a bemeneti fájl típusokról az OpenLayers dokumentáció szolgál részletes információval. Az OpenLayers a JavaScript API-n keresztül valósítja meg a térkép interaktivitását. Hasonlít a Google Maps szolgáltatásához azzal a nagy különbséggel, hogy az OpenLayers nyílt forráskódú, míg a Google nem. Az első verziót az amerikai MetaCarta cég készítette el 2006-ban, 2007 vége óta pedig – a MapServerhez hasonlóan – az OSGeo felügyeli a programot. (Open Source Geospatial Foundation, 2007; OpenLayers Dokumentáció, 2008; Gede, 2016)

Az OpenLayers API kódjait használhatjuk linkelve, vagy a saját szerverünkre letöltve. Hivatkozás esetén mindig a legfrissebb kódcsomagot érjük el, nem kell a szerveren lévő letöltésünket állandóan frissíteni. Viszont, ha a kódokat tartalmazó weboldal, a `http://openlayers.org` valamilyen hiba miatt – például a hoszt szerver elérhetetlensége, a helyi internet szolgáltatás szünetelése – leáll, akkor ez az általunk készített térképalkalmazást is használhatatlanná teszi (Gede, 2016). A hivatkozás a HTML fájl fejlécében történik az alábbi kóddal:

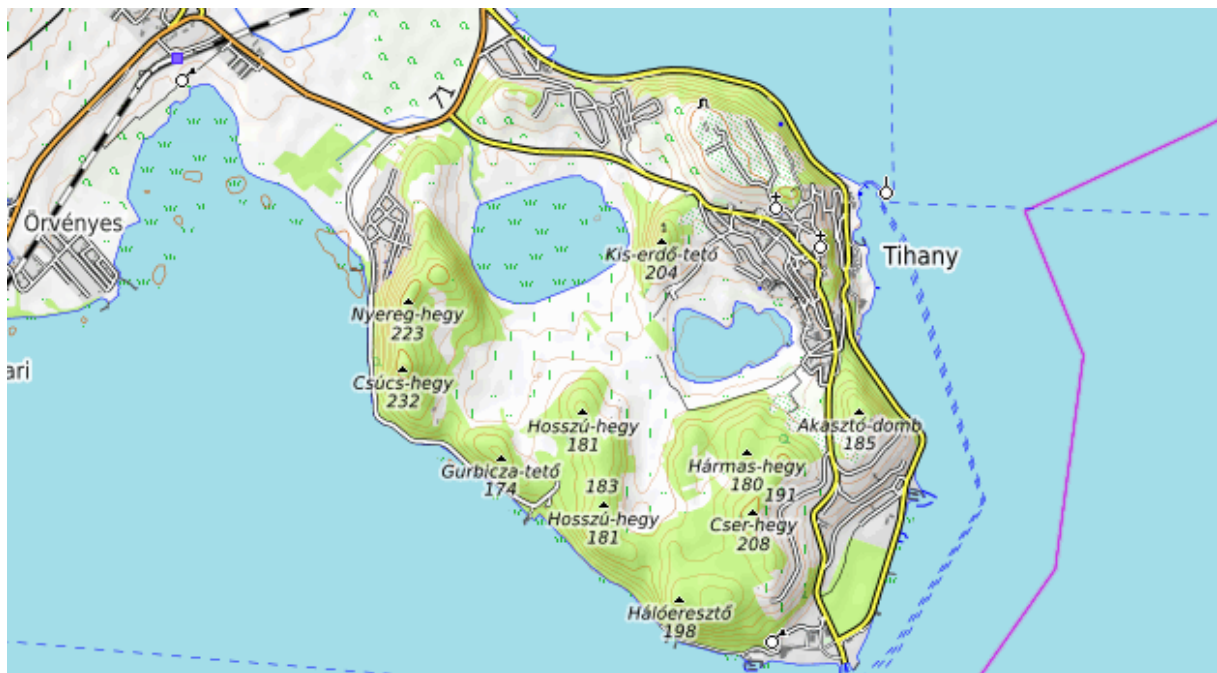
```
<script src="//openlayers.org/api/OpenLayers.js"
type="text/javascript"></script>
```

Az első és legfontosabb lépés a térkép kialakítása. A fejlécben meghivatkozott OpenLayers API mellett definiálnunk kell a térképünk helyét a `<div>` elem használatával. A térképet tartalmazó változó meghívásakor erre a `<div>` elemre kell hivatkoznunk, hogy az OpenLayers elkészítse a térképi felületet. A térkép helyének meghatározása után létre kell hoznunk a térképet tartalmazó objektumot, amelyet az `OpenLayers.Map` konstruktor segítségével végezhetünk el. Az `OpenLayers.Map` legalább egy paramétert kap, amely vagy egy önálló HTML objektum, vagy annak azonosítója lehet. Ezt határoztuk meg korábban a `<div>` elemben az ID attribútummal. Egy további paraméterrel különféle beállítások adhatók meg, amely jelen esetben a vetület. Az alábbi kód – amely része a saját HTML fájlomnak is – készíti el a térkép objektumot, és létrehoz egy OSM réteget, amelyen az `OpenStreetMap` jelenik meg. Ennek a vetülete a Spherical Mercator (EPSG:3857), ez a réteg csak ebben a vetületben jeleníthető meg. (Gede, 2016)

```
var map = new OpenLayers.Map('map', {projection:'EPSG:3857'});
```

```
map.addLayer(new OpenLayers.Layer.OSM());
```

A fenti kóddal definiáltuk az alap OpenStreetMap réteget. Munkám során elkészítettem két másik térképi háttérrel is – az OpenCycleMap-et és az OpenTopoMap-et. Azért választottam az OpenCycleMap térképet, mert elűt az OpenStreetMap-től a színvilága, illetve szintvonalas domborzatábrázolás, valamint kerékpáros tematika van rajta. Az OpenTopoMap nem egy szokványos háttértérkép, melyen szintén van szintvonalas domborzatábrázolás, illetve a szintkülönbség jobb érzékeltetéséért domborzatárnyékolás (summer) is található rajta (8. ábra). Fontos megjegyezni, hogy az alaprétegek közül egyszerre csak egy lehet aktív, melyet a rétegkezelőben választhatunk ki.



8. ábra: OpenTopoMap, mint háttértérkép

Az OpenLayers térképek másik fontos osztálya a LAYER. Ennek használatával fedvényeket hozhatunk létre, amelyek a rétegkezelőben tetszés szerint ki-bekapcsolhatók. Ezekből egyszerre többet is tarthatunk aktív állapotban. Az overlay rétegekhez szükségünk lesz a saját adatforrásainkra. Az előző alfejezetekben tárgyalt, mapfile-ban tárolt információk és beállítások alapján ábrázolhatók saját adataink. Ehhez egy WMS réteget kell létrehoznunk és hozzáadunk a térképhez. Az általam készített turistautak térképi megjelenítését az alábbi kód végzi:

```
var wms=new OpenLayers.Layer.WMS('Turistautak',  
'http://mercator.elte.hu/cgi-bin/mapserv?map=/home/hal2013/fakanpeter/public_html/diplomamu  
nka/jelzett_csf.map',{  
  layers: ['jelzettutak'],  
  isBaseLayer: false,
```

```
transparent: true },  
{singleTile: false });  
map.addLayer(wms);
```

Látható, hogy ezt a réteget is egy konstruktor építi fel, a többi pedig a WMS réteg beállítása. A konstruktornak paraméterként adjuk meg a rétegkezelőben megjelenő nevet, és a mapfile elérési útvonalát. Ezután kapcsos zárójelen belül (JavaScript objektumként) következnek a rétegbeállítások. A `layers` a mapfile-on belül definiált 'jelzettutak' nevű csoportba tartozó rétegeket jeleníti meg. Az `isBaseLayer` beállításával dönthetjük el, hogy alapréteg (`true`), vagy fedvény (`false`) legyen az ábrázolt adat. Minden egyes overlay réteg külön változóval határozandó meg. A kód végén pedig hozzáadjuk az új 'wms' réteget a térképhez. Több overlay réteg is hozzáadható egy paranccsal, ekkor szögletes zárójelben kell hivatkoznunk a rétegek neveire.

```
map.addLayers([wms, csp, kil, mue, muz, tan, ter, wms2]);
```

A munka során létrehoztam a turistautak, elágazások, kilátóhelyek, múzeumok, műemlékek, tanösvény állomások és természeti értékek nevű overlay rétegeket.

A rétegek megjelenítése után szükséges még beállítani, hogy az oldal betöltődésekor melyik területet mutassa a térkép. A cél az, hogy egyből azt a területet lássuk, ahol a tematikát lehet böngészni. Ehhez a `zoomToExtent` függvényt kell úgy paraméterezni, hogy megadjuk a használt vetületben (Spherical Mercator) a befoglaló téglalap négy oldalának megfelelő koordinátákat. A kód lefutása után a térképünk középpontja a meghatározott befoglaló téglalap közepére kerül. A `zoomToExtent` függvény az `OpenLayers.Map` objektum része:

```
map.zoomToExtent(new  
OpenLayers.Bounds(1987000, 5925000, 1992000, 5930000));
```

5.4 Interaktivitást biztosító funkciók

Az OpenLayers megjelenítővel nemcsak adatainkat tudjuk megjeleníteni böngészhető térképeken, hanem interaktívvá is tudjuk tenni azokat a felhasználó számára. Több lehetőség adott, de mivel WMS rétegeket használtam – ezáltal az adatok raszteres kimenetben jelennek meg – ezért csak azt fogom részletezni, amelyeket a WMS támogat. A másik lehetőség, hogy az adatokat vektoros rétegen keresztül ábrázoljuk, és az OpenLayers `SelectFeature` eszközével tesszük az egyes elemeket kijelölhetővé.

A térképem alapvető interaktivitását a `WMSGetFeatureInfo` eszköz fogja biztosítani. Ez a vezérlő egy WMS kérést küld a szervernek, melyre válaszként azt várja, hogy a kattintás pontjában milyen elem található. A válasz háromféle formátumban érkezik beállításától függően: lehet sima szöveg, formázott HTML fájl vagy GML (Geography Markup Language) adat (Gede, 2016).

A helyes működéshez szükséges a `mapfile`-t is konfigurálni a WMS szolgáltatás igénybevételéhez. A MapServer 6.0 verziójától kezdve a következő kódsorokat kell beilleszteni:

```
WEB
METADATA
    "wms_enable_request" "*"
    "wms_feature_info_mime_type" "text/plain"
    "wms_feature_info_mime_type" "text/html"
END
END
```

A `LAYER` objektumon belül is be kell állítanunk az alábbi sorokat, hogy a kiválasztott formátumú válasz helyes legyen:

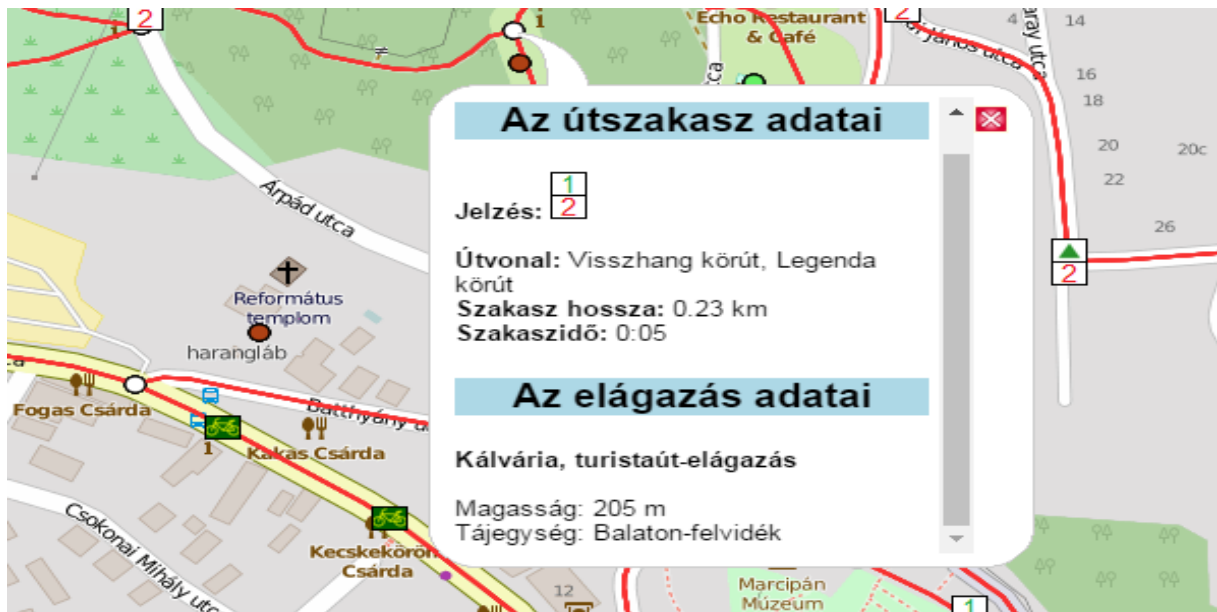
```
METADATA
    "wms_enable_request" "*"
    "gml_include_items" "all"
    "wms_include_items" "all"
END
```

A beállítások megadásával tudunk fogadni `text/plain`, `text/html` és `application/vnd.ogc.gml` formátumú válaszokat. A definiálás a konstruktoron belül az `infoFormat` használatával lehetséges. Szükséges még a `mapfile`-ban egy `TEMPLATE` objektumot létrehozni, ugyanis e nélkül semelyik formátum sem működik. A HTML formátumhoz ezzel egy sablont írunk, így az információ az előre eltervezett módon jelenhet meg. A fájlt bármilyen HTML kóddal formázhatjuk.

A funkció működéséhez az `OpenLayers.Control.WMSGetFeatureInfo` vezérlőt kell paraméterezni. Szükséges ehhez egy külön WMS réteget létrehozni a funkcióhoz. Egy helyesen működő – a térképem részét képező – `WMSGetFeatureInfo` kódja alább látható.

A kapott eredményt a következő két ábra mutatja be (9. ábra, 10. ábra).

```
var gf=new OpenLayers.Control.WMSGetFeatureInfo({
  infoFormat: "text/html"
  layers: [wms2],
  maxFeatures: 1,
  eventListeners:{
    getfeatureinfo: function(e){
      if (e.text!=""){
        while (map.popups.length>0)
          map.popups[0].destroy();
        var p=new OpenLayers.Popup.FramedCloud
          ('infoPopup',map.getLonLatFromPixel(e.xy),
          null,e.text,null,true);
        p.maxSize=new OpenLayers.Size(700,700);
        map.addPopup(p);}}});
map.addControl(gf);
gf.activate();
```



9. ábra: A szövegbuborékba került adatok a `WMSGetFeatureInfo` segítségével

A fenti kóddal nem csak a `WMSGetFeatureInfo` eszközt definiáltuk, hanem egy felugró popup ablakot is, amelyben a HTML nyelven formázott információt jelenítjük meg. A `gf` és `p` változók a `WMSGetFeatureInfo` vezérlőt és a popup ablakot határozzák meg. A felugró ablak maximum méretét a `p.maxSize=new OpenLayers.Size` segítségével adjuk meg. A `layers: [wms2]` azt jelenti, hogy a 'wms2' nevű WMS rétegre lesz érvényes a beállítás. Két WMS réteg lett létrehozva, egyik az adatok megjelenítésére szolgál, a

másik pedig a `WMSGetFeatureInfo` működését biztosítja. Utóbbi nem tartalmaz tematikát, és mivel fontos az interaktivitás szempontjából, ezért nem kapcsolható a rétegválasztóban. A `maxFeatures` paraméter beállításával meghatározhatjuk, hogy hány elemről érkezzen információ, ha a kattintás helyén több lekérdezhető réteg van. Amennyiben nincs adat, a popup ablak nem jön elő. A kód végén mind a popup ablakot, mind a `WMSGetFeatureInfo`-t hozzá kell adni a térképhez, valamint az utóbbit aktiválni is kell.



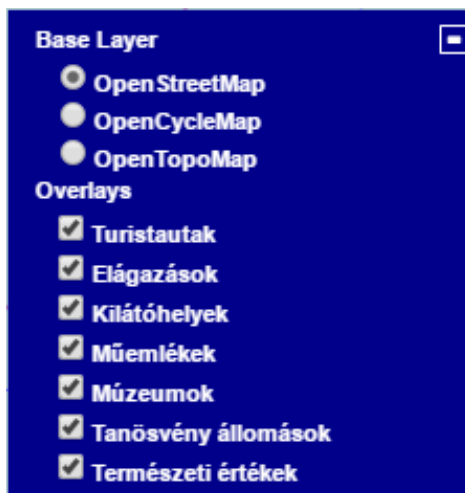
10. ábra: A szövegbuborékba került adatok a látványosság nevű lekérdezhető rétegről

A korábban már említett rétegkezelőt is meg kell hívnunk a hozzá tartozó konstruktorral. A rétegválasztó elkészítésével emelni lehet a térkép interaktivitását, mivel így a felhasználó döntheti el, hogy mely alapréteget és tematikus rétegeket szeretné látni (11. ábra). A rétegkezelő control az alábbi kóddal építhető fel. Ha másképp nem rendelkezünk, a rétegválasztó a térkép jobb felső sarkába kerül automatikusan.

```
var layerSwitcher = new OpenLayers.Control.LayerSwitcher();  
map.addControl(layerSwitcher);
```

A kezelőszervet, a definiálás után a `map.addControl` segítségével adhatjuk hozzá a térképhez. Ezt a folyamatot az alábbi kódrészlet végzi:

```
map.addControl(new OpenLayers.Control.LayerSwitcher());
```



11. ábra: A rétegválasztó kezelőfelület

A rétegek alapértelmezettként megjelennek a rétegkezelőben, attól függően, hogy alapréteg (baselayer) vagy fedvény (overlay) típusú. Ha azt szeretnénk, hogy ne legyen automatikusan aktív valamelyik overlay réteg, akkor a `setVisibility` tulajdonságot kell beállítanunk `false` értékre. Ha nem szeretnénk a réteget megjeleníteni, mert például azt csak a `WMSGetFeatureInfo` használatához hoztuk létre, a `displayInLayerSwitcher` beállítást kell `false` értékre állítani. Mindkét paraméter az `OpenLayers.Layer` objektum tulajdonsága. Ezt a definíciót kapcsos zárójelben kell megadni az új WMS réteg beállításain belül, különben nem fog működni.

```
layer.setVisibility(false);
{displayInLayerSwitcher: false}
```

A térképet böngészve érdekes lehet a kurzor aktuális pozíciójának koordinátáit kiírni. Ehhez az `OpenLayers.Control.MousePosition` vezérlőt kell meghívunk. Alap esetben a használt vetület szerint írja ki a koordinátákat, viszont megfelelő beállítás esetén más vetület koordinátái is megjeleníthetők.

```
var mousePositionCtrl = new OpenLayers.Control.MousePosition({
  prefix: 'Pozíció: ',
  separator: ' | ',
  numDigits: 5,
  emptyString: 'A kurzor nincs a térképen.',
  displayProjection: "EPSG:4326"});
map.addControl(mousePositionCtrl);
mousePositionCtrl.activate;
```

Lehetőségünk van `prefix` vagy `suffix` megadására, azaz, hogy milyen szöveg jelenjen meg a koordináta pár előtt vagy után. A `separator` a két koordináta közötti elválasztó karaktert adja meg. Mivel ez egy `{string}` típusú beállítás, ezért bármilyen szöveget beírhatunk. A `numDigits: {Integer}` paraméter azt adja meg, hogy hány tizedes értékig íród-

jon ki az érték. Az `emptyString: {string}` tulajdonság arra szolgál, hogy mit írjon ki a koordináták helyére, amennyiben a kurzor nem a térképen található. A `displayProjection` megadja, hogy mely vetület koordináta értékét írassuk ki. A WGS-84 megfelelője az EPSG:4326. A `control` hozzá kell adni a térképhez, és aktiválni kell.

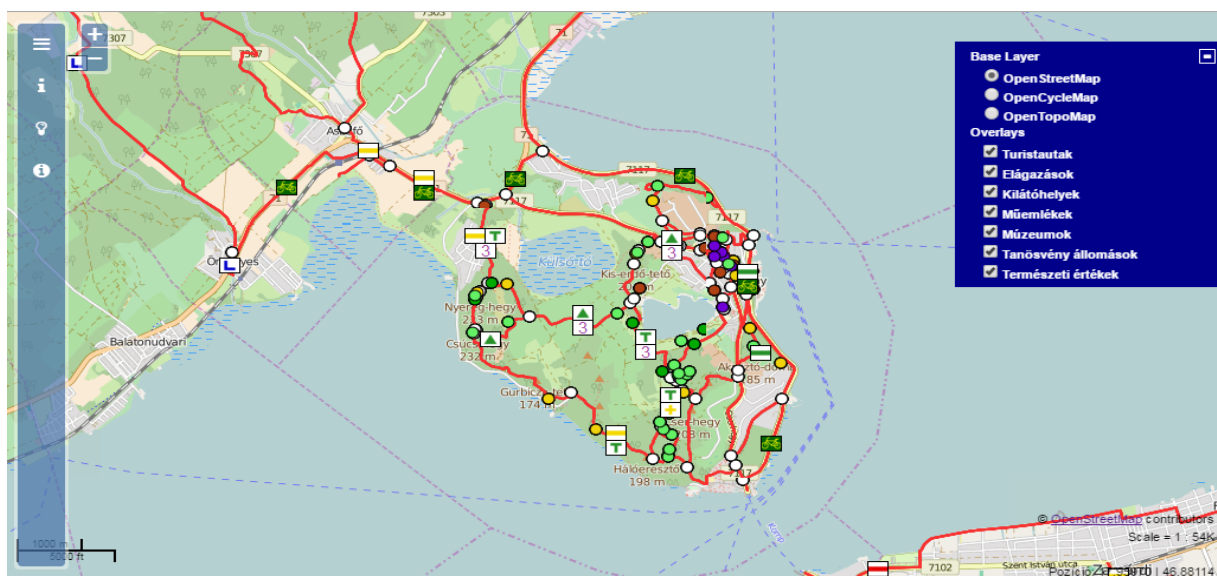
Interaktívan jeleníthetjük meg a térképünk aktuális méretarányát számértékkel, ha egy `OpenLayers.Control.Scale` objektumot is hozzáadunk. Alaphelyzetben a méretarány a térkép jobb alsó sarkába kerül. A változó deklarálása után ezt is a térképünkhöz kell adni.

```
var scale = new OpenLayers.Control.Scale();
map.addControl(scale);
```

Lehetőség van az aktuális méretarányt mértékléccel is szemléltetni. A mértékléc létrehozásához az `OpenLayers.Control.ScaleLine` vezérlőt kell használnunk. Alap esetben a mértékléc a térkép bal alsó sarkába kerül. Ez a `control` is hozzáadandó a térképhez.

```
var scaleline = new OpenLayers.Control.ScaleLine();
map.addControl(scaleline);
```

Mivel a térképem egész oldalas az okostelefonok böngészőjében való jó megjelenítés érdekében, ezért importáltam egy előre elkészített ki-becsukható panelt a térkép bal oldalára. A `Sidebar-v2` névre hallgató panelt az MIT licenc engedélyével használtam fel, feltétele, hogy a licencet és a forrást fel kell tüntetni az oldalon. Az interaktív oldalpanelt CSS fájlban formázhatjuk, és HTML segítségével ágyazhatjuk be a weboldalba (Bieniek, 2013). Ez tartalmaz egy rövid használati utasítást a webtérképhez, a jelmagyarázatot, melyet a `MapServer` generál, egy keresődobozt, mellyel az általam készített tematikában kereshetünk, a licenc és a forrás megnevezését, valamint, hogy a térképi alkalmazás jelen diplomadolgozat mellékleteként készült. A kész interaktív térképes weboldal kinézete alább látható (12. ábra).



12. ábra: Az interaktív térképes weboldal kinézete

6. ÁBRÁZOLÁSI MEGOLDÁSOK- ÉS KORLÁTAIK

6.1 Turistautak ábrázolása

Miután láttuk, hogyan kell paraméterezni a mapfile-t az adataink térképi ábrázolásához, illetve megnéztük, hogy hogyan működik az OpenLayers keretrendszer, amely megjeleníti a MapServer által küldött adatokat, említést kell tennünk a munkafolyamat során fellépő ábrázolási megoldásokról- és korlátaikról.

A turistatérkép olyan közepes méretarányú topográfiai térkép, amely közvetlen terepi tájékozódásra alkalmas, valamint a túrázás szempontjából alapvető fontosságú információk, jelzések és tereptárgyak közvetlenül leolvashatók. Célja a terep részletes bemutatása, valamint az idegenforgalom szempontjából lényeges jelek és objektumok kiemelt ábrázolása (ITM-ATSZT2E; Faragó, 2014).

A turistautakat és jelzéseket többféle módszerrel is ábrázolhatjuk a térképen. A jelzett utakat, turistautakat térképészeti szempontból két külön részre bonthatjuk. A folytonosságot a vonalas elemekkel, az egyes utak elhatároltságát pedig a vonalra vonatkoztatott névvel vagy jellel szemléltetjük. A turistautak ábrázolása kétféleképpen lehetséges. A vonalakat megjeleníthetjük a terepi jelzéstől függetlenül egy kiemelt térképi színnel – általában vörös (13. ábra) –, a turistajelzéseket pedig jelölhetjük a jelzés színének rövidítésével vagy a terepi valós jelzés feltüntetésével.



13. ábra: Ábrázolás kiemelt térképi színnel és terepi valós jelzésekkel

Másik lehetőség, hogy a turistautakat a jelzés színével megegyező vonallal ábrázoljuk, a turistajelzéseket pedig az előbb ismertetett két módszer valamelyikével illusztráljuk (14. ábra). Olyan megoldás is lehetséges, hogy a turistautakat alapvetően egy kiemelt színű vonallal nyomjuk felül, de például az országos szinten nagy jelentőségű túraútvonalakat a terepi jelzés színével jelenítjük meg (15. ábra). Utóbbi esetre jó példa az Országos Kéktúra nyomvonala, vagy a Mária út egyedi lila jelzése. Az alábbi fejezetben a turistautak MapServer-en és OpenLayers-en keresztüli ábrázolási lehetőségeit és korlátait fogom vizsgálni, a lehető legjobb mellett állást foglalni és a mintaterület turistaútjait megjeleníteni. (Faragó, 2014)



14. ábra: Ábrázolás a jelzés színével megegyező szín használatával



15. ábra: Ábrázolás kiemelt térképi színnel és jelentős túraútvonal kiemelésével

6.1.1 Ábrázolás kiemelt térképi színnel

Az adatforrás szerkezete ebben az esetben úgy néz ki, hogy a shapefile-ban a közös nyomvonalon futó jelzések egy vonalas elemként vannak tárolva, ahol a *[name]* mező tartalmazza a turistajelzéseket vesszővel elválasztva. A kiemelt térképi színnel történő ábrázolásakor a turistautakat egy 250 50 50 RGB színkódú – tehát élénkpiros –, valamint 2 pixel szélességű vonallal jelenítettem meg. Mivel ezek vonalas elemek, ezért a mapfile-ban egy line típusú LAYER objektumot kellett definiálnom. Ez a későbbi megoldások során is így lesz meghatározva a mapfile-ban. Az adatok elérési útvonala, a vetület beállítása után a turistautak hibátlanul megjelennek a térképen. A turistautak megjelenítéséről gondoskodó kód a térképemhez szükséges mapfile-ban alább látható:

```
LAYER
  NAME jelzett_utak
  TYPE line
  DATA „jelzesek_2016.shp”
  STATUS DEFAULT
  PROJECTION
    „init=epsg:4326” # WGS-84 latlon
  END
  CLASS
  STYLE
    WIDTH 2
    COLOR 250 50 50
  END
END
END
```

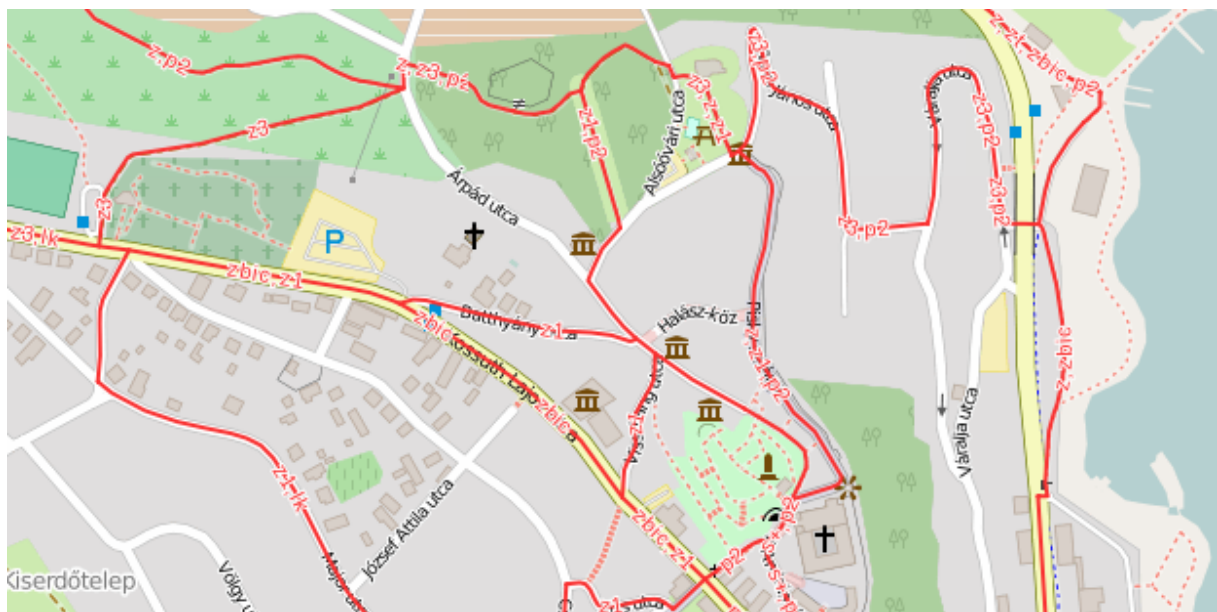
A következő lépés a terepi jelzések szemléltetése a térképen. A fejezet elején ismerttettem ennek két lehetséges kimenetét. Most ezeknek a megvalósítását fogom bemutatni. Az első eset, hogy a jelzéseket a színek nagy kezdőbetűivel illetve a hozzátartozó geometriai ábrát helyettesítő karakterrel jelölöm. Erre azért van szükség, mert a shapefile-ban a *{string}* típusú adatmezőben csak betűket, számokat és speciális karaktereket tudunk tárolni. Így lesz például a sárga háromszög turistajelzésből S3, vagy a kék négyzetből K4. A teljes lista a rövidítésekkel az interneten megtalálható (Gede, 2016; OpenStreetMap, 2016). A színek kezdőbetűinek kiírása a mapfile-on belül a LABEL objektum definiálásával történik. Mivel a jelzés vonalra vonatkozó elem, ezért ennek is külön réteget kell létrehozunk a mapfile-ban, amely *annotation* típusú. A LABELITEM értéke a 'NAME', tehát az adatforrás így nevezett mezője alapján feliratozunk. Az OFFSET paraméter beállításával eltolhatjuk a szöveget a vonaltól, ám ezt a MapServernek csak bizonyos verziói teszik lehetővé, és sajnos az sem megbízhatóan működik. Így ebben az esetben, a karakterek a vonal futását követik és a vonalra helyeződnek.

Ez a megjelenítés az internetes térképek esetén kevésbé előnyös, mivel a vonalra írt karaktereket nehezebben lehet felismerni, mintha mellette lennének (16. ábra). Az alábbi kód elkészítésével lehet formázni a szöveget és megjeleníteni azt a térképen:

```

LABELITEM 'NAME'
CLASS
  LABEL
    TYPE truetype
    FONT sans
    SIZE 8
    ANGLE follow
    COLOR 250 50 50
    OUTLINECOLOR 255 255 255
    OUTLINEWIDTH 1
  END #label
END #class

```



16. ábra: Ábrázolás kiemelt térképi színnel és a jelzés kezdőbetűjével

A másik módszer, hogy a terepi jelzéssel ábrázoljuk a turistautak jelzéseit. Ekkor egy külső forrásból származó piktogramot jelenítünk meg, ami a vonalas elemre vonatkozik. Az adatforrás [*img_url*] mezőjében a fent említett PHP script elérési útvonalát hivatkozzuk meg, amely legenerálja a szimbólumokat, a térképen pedig a mapfile-ban lévő definíció szerint jelenik meg. Ha a turistaúton három vagy annál több jelzés halad, akkor a piktogramok két oszlopban jelennek meg. Sokkal szemléletesebb ez a módszer, mivel a jelzések egyértelműen elkülöníthetőek és jobban értelmezhetőek, mintha csak a szín kezdőbetűje lenne látható (17. ábra). Ez a megoldás az alábbi kóddal működik (Gede, 2016):

```

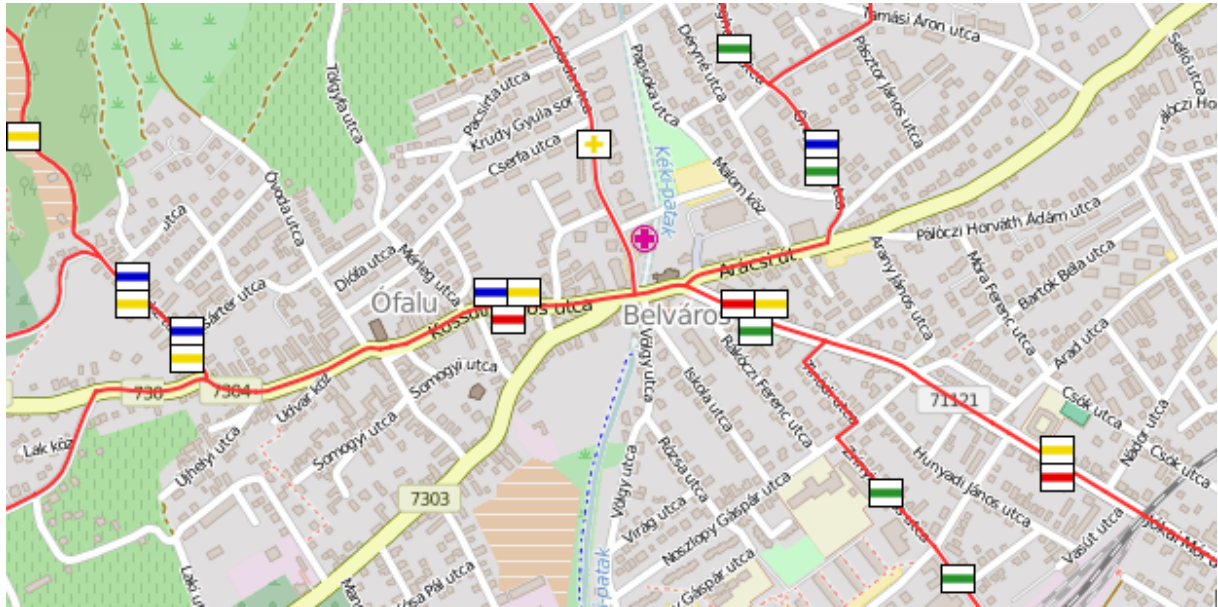
CLASS
  TEXT " "
  STYLE

```

```

SYMBOL [img_url]
END
LABEL
MINDISTANCE 50
PARTIALS false
END #label
END #class

```

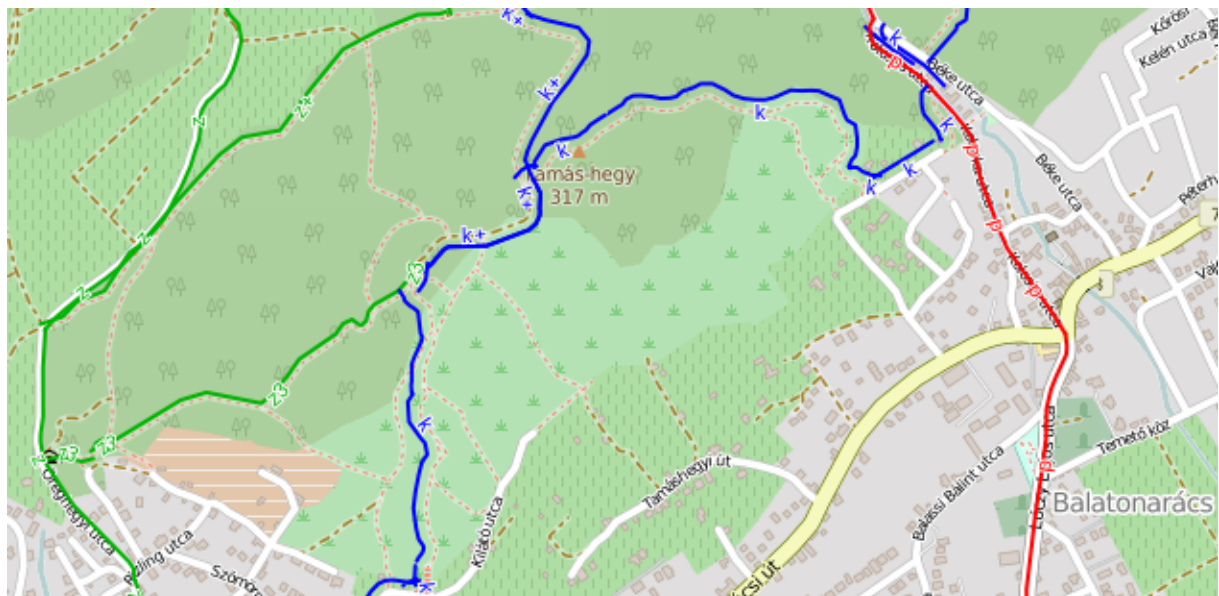


17. ábra: Ábrázolás kiemelt térképi színnel és a terepi valós jelzés feltüntetésével

6.1.2 Ábrázolás a jelzés színével azonos szín használatával

A forrásként használt shapefile ebben az esetben minden egyes turistaút külön vonalas elemként tárol. Tehát, ott ahol több turistaút fut közös nyomvonalon, ott többször szerepel ugyanaz a szakasz a shapefile-ban. A [name] adatmező ekkor csak egy jelzésnek a kezdőbetűjét és esetleges speciális karakterét tárolja. A turistaútak megjelenítésére szolgáló vonalas elemeket itt is az előző alfejezetben taglalt leírás szerint hoztam létre, azzal a különbséggel, hogy itt a vonal színe a terepi jelzés színével megegyező. Az adott színhez rendelt megfelelő vonalas elem kiválasztásához a mapfile-ban az EXPRESSION feltételt kell megszabnunk. Ezt úgy kell definiálnunk, hogy a kiválasztás az adott szín a jelzések hierarchiájának összes elemét – tehát a sávjelzéseket, a kereszt jelzéseket és a formajelzéseket – tartalmazza. A következő kódrészlet a zöld színű jelzéseket választja ki, és a STYLE alapján zöld színnel jeleníti meg ezeket a vonalas objektumokat, illetve 2 pixellel tolja el az eredeti helyétől (18. ábra):

```
CLASS
NAME „zold”
EXPRESSION /^z/
STYLE
WIDTH 2
COLOR 0 180 0
OFFSET -2 -99
END
```

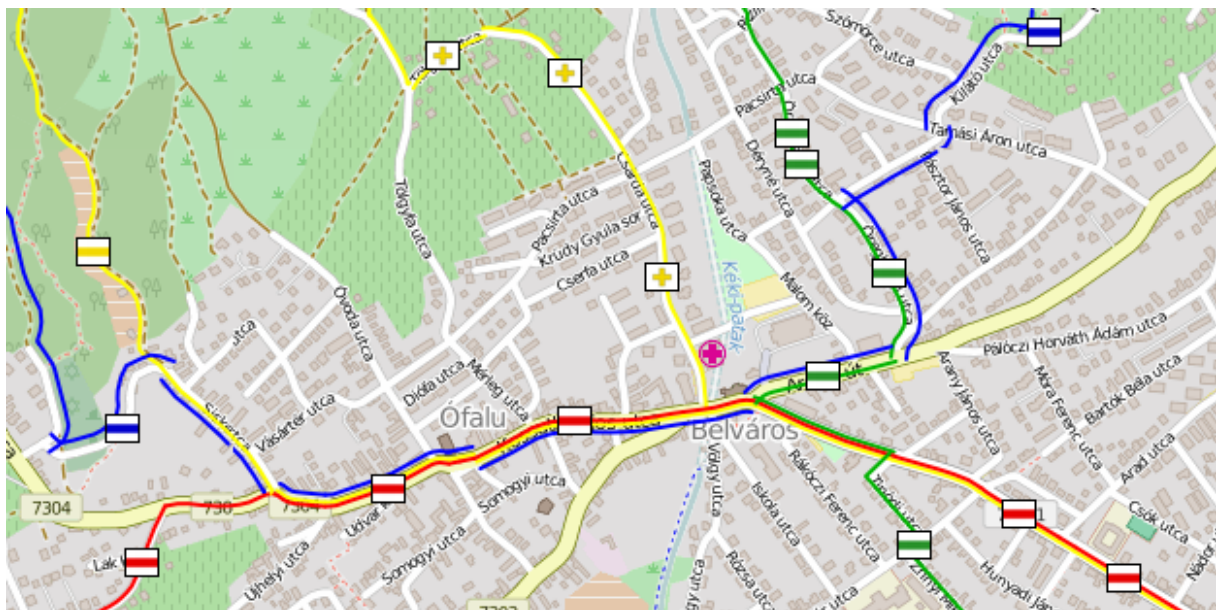


18. ábra: Ábrázolás a jelzés színével megegyező színnel és a jelzés színének kezdőbetűjével

Ezt a szelekciót minden egyes színre el kell végeznünk. Magyarország esetében ez legalább négyféle színt – kék, piros, sárga és zöld – jelent. Létezhetnek még tematikus útvonalak is,

amelyek további színeket vehetnek fel. Minél többféle színű turistautat jelenítünk meg ezzel a módszerrel, annál nehezebb ezt megvalósítani. A különböző színű turistautakat a közös szakaszon egymáshoz képest el kell tolni, a jó olvashatóság szempontja miatt. Ezt az OFFSET[x][y] paraméterezésével tehetjük meg. Az [y = -99] beállítás az eredeti elemhez képest merőlegesen tolja el a vonalat [x] pixel távolságra. Mivel a turistautakat 2 pixel széles vonallal jelenítjük meg, ezért az [x] értékét 2 pixellel kell növelnünk, hogy egymás mellett legyenek szorosan a vonalak. Elméletben idáig ez jó megoldás, sajnos azonban a MapServer a töréspontoknál ismeretlen okok miatt felcseréli a színek futását, emiatt nem konzekvens a megjelenítés sok esetben. Egymástól távolra eső színek esetében nagy hézag keletkezik a két turistaút között, mivel nem tudható, hogy melyik színű utak milyen arányban futnak közös nyomvonalon (18. ábra, 19. ábra).

A turistajelzések piktogramjainak elérési útvonalát szintén az [img_url] mező tartalmazza. A szimbólumok meghívása a mapfile-on belül ugyanúgy történik, mint az előző esetben. Az alapvető különbség itt az, hogy nem a PHP script generálja a jelzéseket, hanem a szerveren megtalálható PNG formátumú képeket használja fel a MapServer. A szimbólumok megjelenítésénél szintén probléma, hogy a MapServer csak egy adott pontba hajlandó kihelyezni a PNG ábrát, azaz, ha közös nyomvonalon több turistaút fut, ott is csak az egyik jelzés lesz látható (19. ábra). A terepi valós jellel való ábrázolás tehát működik itt is, viszont mivel az OFFSET működése kartográfiai szempontból kifogásolható, illetve a jelzések általános megjelenése nem kielégítő, így sajnos ez a módszer nem az igazi megoldás.



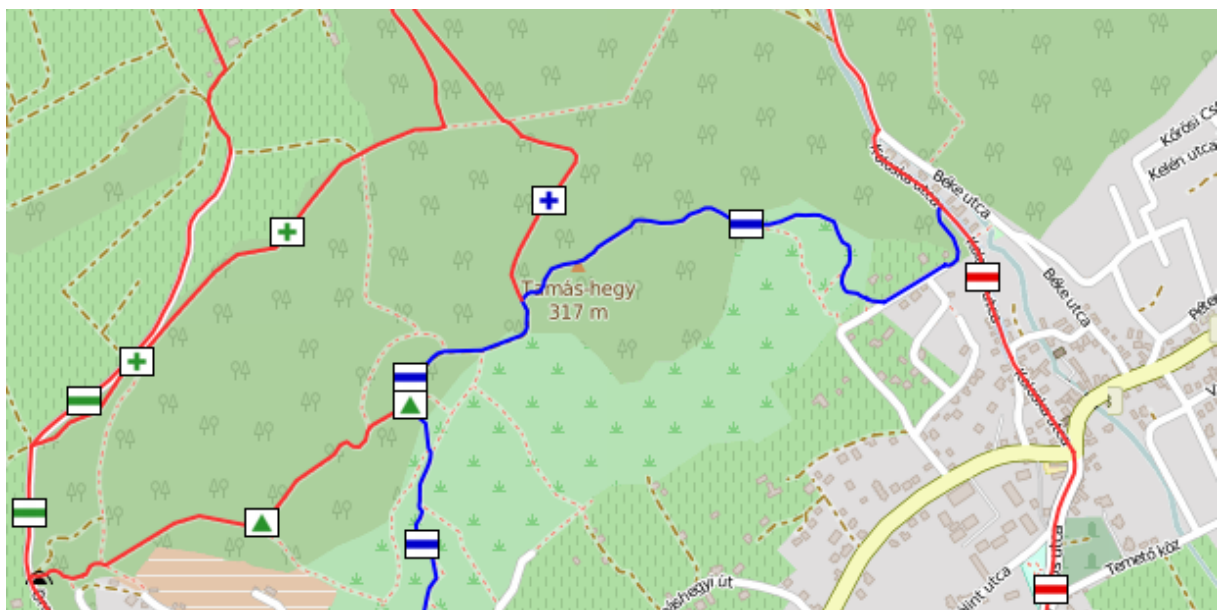
19. ábra: Ábrázolás a jelzés színével megegyező színnel és a terepi valós jellel

6.1.3 Ábrázolás kiemelt térképi színnel és jelentős útvonallal

Ebben a megoldásban az adatforrás ugyanúgy néz ki, ahogy a 6.1.1 alfejezetben meg lett határozva. A kiemelt térképi szín ismét a 250 50 50 RGB színkódú élénkpiros lesz, a jelentős túraútvonal színe pedig a 0 0 255 RGB színkódú kék. Jelen esetben az utóbbi a mintaterületen áthaladó Balatoni Kék túraútvonal lesz, de ez természetesen használható akár az Országos Kéktúra vagy bármi más ábrázolására. A kiválasztás folyamata hasonló a fenti esethez, azzal a különbséggel, hogy itt csak a kék sáv attribútumot tartalmazó vonalak lesznek kék színűek. Minden más elem, – ami nem került kiválasztásra – az a másik osztályba kerül, amelyek élénkpiros színnel jelennek meg (20. ábra). Ezt a folyamatot jelenti az alábbi kódrészlet:

```
CLASS
  EXPRESSION / (^|, )k($|, ) /
END
CLASS
  STYLE
    WIDTH 2
    COLOR 250 50 50
  END
END
```

A feliratozást ebben az esetben is ugyanúgy kell elvégezni, mint a fenti esetekben. Javasolt itt is a turistajelzések piktogramjait megjeleníteni a PHP scripten keresztül, mert ez látványos és sokkal informatívabb, mint egy szín kezdőbetűje (20. ábra). Mivel ebben a megoldásban nem okoz gondot az OFFSET, és a szimbólumok is szépen megjelennek a turista-utak egy vonalas elemmel való ábrázolása miatt, ezt találok a legjobb kivitelnek.



20. ábra: Ábrázolás kiemelt térképi színnel és jelentős útvonallal

6.2 A pontszerű objektumok ábrázolása

A pontszerű objektumok térképi ábrázolása kevésbé bonyolult, és nincs is ennyi féle variáció, amelyekből válogatni lehet. Ezeknek a megjelenítési beállításait szintén a mapfile-ban kell definiálni, csakúgy, mint a vonalas elemekét. A MapServer ebből az információból készíti el a térképet. A pontszerű elemeket kétféle rétegen is elhelyezhetjük, mely lehet *point* vagy *annotation* típusú. Az alábbiakban megvizsgálom, melyik típus milyen előnyökkel és hátrányokkal bír, illetve ismertetem az általam létrehozott pontszerű tematikus rétegek szerkezetét, felépítését.

A *point* típusú réteg alkalmazása esetén a térképünk minden egyes pontszerű objektumot kirajzol, kivéve, ha beállítunk rá méretarányfüggő megjelenítést. Hasonlóan épül fel, mint a korábbiakban ismertetett *line* típusú réteg. A különbség, hogy még a réteg előtt deklarálni kell a megjelenítendő pontszerű jelet a SYMBOL használatával. Az előre elkészített szimbólumokra a réteg készítése során a CLASS/STYLE szinten kell hivatkozni, hogy a MapServer tudja, melyik réteg elemeihez melyik szimbólumot kell hozzárendelni. Külső forrásból származó kép is megjeleníthető, ekkor a SYMBOL típusa *pixmap*, illetve kötelező megadni az elérési útvonalat. A TOLERANCE beállításával megadhatjuk, hogy mekkora legyen a kattintás pontossága. A turistautak csomópontjainak megjelenítését végző kódrészlet itt látható:

```
LAYER
NAME jelzett_utak_elagazas
TYPE point
DATA „csomopont.shp”
STATUS DEFAULT
TEMPLATE „csomopont_temp.html”
TOLERANCE 5
PROJECTION
„init=epsg:4326” # WGS-84 latlon
END
CLASS
STYLE
SIZE 10
COLOR 255 255 255
OUTLINECOLOR 0 0 0
SYMBOL „csomopont”
END
END
END
```

Az *annotation* típusú réteg alapvető különbsége, hogy a pontszerű objektum csak akkor rajzolódik ki, ha vagy kiíratjuk a hozzátartozó névrajzi elemet, és az kifer a térképre, de abban az esetben is megjelenik, ha magát az elemet nem takarja ki másik pontszerű jel.

7. ÖSSZEGZÉS

A diplomamunka célja egy nyílt forráskódú webes térképi alkalmazás elkészítése volt, amely tekinthető az alapszakon készített korábbi szakdolgozatom továbbfejlesztésének is. A témaválasztás során elsősorban erre helyeztem a hangsúlyt, illetve fontosnak éreztem, hogy a mesterképzésen elsajátított tanulmányaim alapján megfelelően tudjak megoldani térinformatikai feladatokat és problémákat.

A diplomadolgozat első felében áttekintettem a Magyarországon leginkább ismert internetes térképeket, illetve kitértem két külföldi – szlovák és lengyel – webtérképes megoldásra is. Megvizsgáltam a térképek ábrázolási módszereit, és interaktivitási lehetőségeit, bemutattam az egyes térképek – általam vélt – pozitív és negatív megoldásait. Kiválasztottam azt az internetes térképi megoldást, amelyhez hasonlóan alkottam meg a saját térképi alkalmazásomat. A következő fejezetek során ismertettem a térkép megtervezésének lépéseit, az adatgyűjtés többféle módszerét illetve az adatok lehetséges formátumait. Ezután leírtam az általam elvégzett adatfeldolgozási eljárást.

Az ötödik fejezetben bemutattam a webes térképi alkalmazás elkészítéséhez használt térinformatikai programcsomagot és függvénykönyvtárat. Az első részben ismertettem a MapServer és a mapfile felhasználási lehetőségeit és tulajdonságait. A mapfile-ban használt kódrészletek beillesztésével szemléltettem annak szerkezetét. A második részben áttekintettem a megjelenítésre használt OpenLayers API felépítését és működését. Végül megvizsgáltam a térkép interaktivitását adó funkciókat, és leírtam azoknak a struktúráját és használatát.

A diplomamunka utolsó fejezetében az ábrázolási módszerekre és azok korlátaira fektettem hangsúlyt. Részletesen ismertettem a megjelenítési lehetőségeket, valamint ábrákkal illusztráltam a fellépő problémákat. Többféle alternatíva közül kiválasztottam azt a megoldást, amelyik a legkevesebb korláttal bír, a térképi alkalmazásomat pedig ezzel építettem fel.

Összességében a bevezetésben megfogalmazott célokat elértem, elkészítettem a webes térképi alkalmazást, amely bárki számára elérhető. Véleményem szerint még nem ismeretek széles körben a MapServer és az OpenLayers által nyújtotta lehetőségek, pedig segítségükkel könnyen változatos térképek készíthetők. A diplomamunkában leírtak és a dokumentációk alkalmazásával azonban rengeteg új ötlet és téma valósítható meg.

Az interaktív webes térképi alkalmazás elérhetősége:

<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>

8. FORRÁSJEGYZÉK

8.1 Irodalomjegyzék

(Bieniek, 2013): Tobias Bieniek: Sidebar-v2

JavaScript alkalmazás, 2013

<https://github.com/Turbo87/sidebar-v2>

<https://github.com/Turbo87/sidebar-v2/blob/master/LICENSE>

(Faragó, 2014): Faragó Imre: Sokrétű térképészet IV.: Mesterséges síkrajzi elemek ábrázolása
– 16. Jelzett utak pp. 524-542

Online jegyzet, ELTE, 2014

(Gede, 2016): Dr. Gede Mátyás

A MapServer használata

Online jegyzet, ELTE, 2016

<http://mercator.elte.hu/~saman/hu/okt/mapserver/>

Az OpenLayers API alapjai

Online jegyzet, ELTE, 2016

<http://mercator.elte.hu/~saman/hu/okt/ol/>

OSM Hiking Routes – XMLparseOSM JavaScript program

JavaScript alkalmazás, 2016

https://github.com/samanbey/OSM_hiking_routes

Turistajelzés csoportok webtérképekhez

Online jegyzet, 2016

<http://mercator.elte.hu/~saman/hu/okt/mapserver/turistajelzesek/>

(ITM-ATSZT2E): Térképszerkesztés- és tervezés 2. Előadás diasorai

Faragó Imre

(Magyar Geocaching Közhasznú Egyesület, 2002-2016):

<http://geocaching.hu>

<http://turistautak.hu>

(Mapový Portál Hiking.sk, 2002-2016):

Turistická Mapa Slovenska

<http://mapy.hiking.sk>

(MapServer Dokumentáció, 2016):

<http://mapserver.org/introduction.html>

(MapServer Mapfile Dokumentáció, 2016):

<http://mapserver.org/mapfile/>

(OpenMaps Alapítvány, 2007-2016):

<http://openmaps.eu>

(OpenLayers Dokumentáció, 2008):

<http://docs.openlayers.org/>

(OpenStreetMap, 2016):

Turistajelzések

<http://turistautak.openstreetmap.hu/jelek.php>

<http://openstreetmap.org>

(OpenStreetMap Dokumentáció, 2016):

https://wiki.openstreetmap.org/wiki/Main_Page

(Open Source Geospatial Foundation, 2007):

<http://www.osgeo.org/node/462>

(Overpass API Dokumentáció, 2016):

<http://overpass-api.de/>

(Sygnatura, 2015): Mapa szlaków turystycznych w górach. Kalkulator i planowanie tras.

Zakład Kartograficzny Sygnatura Witold Czajka

<https://mapa-turystyczna.pl/>

<http://www.sygnatura.com.pl/pl/>

(Varga, 2014): Varga Balázs: Karancs webes turista-útikalauz

Diplomamunka, Budapest, 2014

<http://mercator.elte.hu/~vabqabt/dolgozat/terkep.html>

Az internetes hivatkozások utolsó elérése: 2016. június 5.

8.2 Ábrajegyzék

1. ábra: Karancs webes túra-útikalauz OpenStreetMap alapon (részlet)
<http://mercator.elte.hu/~vabqabt/dolgozat/terkep.html>
Diplomamunka, Varga Balázs, 2014
2. ábra: SHOCart turistatérkép a mapy.hiking.sk weboldalon (részlet)
<http://mapy.hiking.sk/>
Mapový Portál Hiking.sk, 2002-2016; SHOCart spol. s r.o., 1991-2016
3. ábra: Interaktív térkép Google Terrain háttérrel és felugró ablakkal (részlet)
<https://mapa-turystyczna.pl/>;
Zakład Kartograficzny Sygnatura Witold Czajka, 2015
4. ábra: Interaktív térkép Sygnatura turistatérkép háttérrel (részlet)
<https://mapa-turystyczna.pl/>;
Zakład Kartograficzny Sygnatura Witold Czajka, 2015
5. ábra: Turistautak attribútum táblája a Quantum GIS programban
Saját ábra
6. ábra: A kész adatbázis Quantum GIS-ben
Saját ábra
7. ábra: MapServer által készített jelmagyarázat
http://mercator.elte.hu/cgi-bin/mapserv?map=/home/hal2013/fakanpeter/public_html/diplomamunka/jelzett_csf.map&mode=legend
8. ábra: OpenTopoMap, mint háttértérkép
<https://opentopomap.org/#map=7/47.006/18.995>
OpenTopoMap, 2016
9. ábra: A szövegbuborékba került adatok a WMSGetFeatureInfo segítségével
<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>
Saját ábra
10. ábra: A szövegbuborékba került adatok a látványosság nevű lekérdezhető rétegről
<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>
Saját ábra

11. ábra: A rétegválasztó kezelőfelület

<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>

Saját ábra

12. ábra: Az interaktív térképes weboldal kinézete

<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>

Saját ábra

13. ábra: Ábrázolás kiemelt térképi színnel és terepi valós jelzésekkel

Őrség, Göcsej, Kemeneshát turistatérkép (2014), 1:60 000 (részlet)

Cartographia Kiadó, Budapest

14. ábra: Ábrázolás a jelzés színével megegyező szín használatával

TatraPlan – The High Tatra turistatérkép (2014), 1:25 000 (részlet)

Zakład Kartograficzny Sygnatura Witold Czajka; TatraPlan s.r.o.

15. ábra: Ábrázolás kiemelt térképi színnel és jelentős túraútvonal kiemelésével

A Bakony, a Bakonyalja és a Balaton-felvidék kiránduló- és szabadidőtérképe (2010),

1:80 000 / 1: 40 000 (részlet)

Szarvas András – Faragó Imre, Budapest

16. ábra: Ábrázolás kiemelt térképi színnel és a jelzés kezdőbetűjével

http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany_csb.html

Saját ábra

17. ábra: Ábrázolás kiemelt térképi színnel és a terepi valós jelzés feltüntetésével

http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany_csf.html

Saját ábra

18. ábra: Ábrázolás a jelzés színével megegyező színnel és a jelzés színének kezdőbetűjével

http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany_szinesbetu.html

Saját ábra

19. ábra: Ábrázolás a jelzés színével megegyező színnel és a terepi valós jelzéssel

http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany_szinesjel.html

Saját ábra

20. ábra: Ábrázolás kiemelt térképi színnel és jelentős útvonallal

<http://mercator.elte.hu/~fakanpeter/diplomamunka/tihany.html>

Saját ábra

9. KÖSZÖNETNYILVÁNÍTÁS

Az elsők között szeretnék köszönetet mondani témavezetőmnek, Dr. Gede Mátyásnak a webes térképi alkalmazás, a weboldal és a diplomamunka elkészítésében nyújtott folyamatos segítségéért. A munka alatt végig biztatott, utat mutatott a fellépő problémák és nehézségek megoldása során.

Köszönet illeti Kiss Tündét, az ELTE Térképtudományi és Geoinformatikai Tanszék térképtárosát, aki segített a megfelelő térképi források felkutatásában.

Köszönettel tartozom Faragó Imrének, aki rendelkezésemre bocsátott az általa készített térképek közül több példányt is, amelyek segítségével kialakíthattam a jelzett útvonalak különböző webes térképi megjelenítését.

További köszönet illeti az ELTE Térképtudományi és Geoinformatikai Tanszék dolgozóit, akikhez bizalommal fordulhattam bármely egyéb felmerülő kérdéssel, problémával.

Szeretném továbbá megköszönni családomnak, közeli barátaimnak és ismerőseimnek, akik a diplomamunka elkészítése során folyamatosan támogattak és segédkeztek.

10. NYILATKOZAT

Alulírott, Fakan Péter Bálint (Neptun-kód: KJF256) nyilatkozom, A Tihanyi-félsziget interaktív webes térképe című diplomamunka szerzője fegyelmi felelősségem tudatában kijelentem, hogy dolgozatom önálló munkám eredménye, saját szellemi termékem, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel. A diplomadolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be. A diplomamunkában felhasznált, szerzői joggal védett anyagokra vonatkozó engedély a mellékletben megtalálható.

A témavezető által elfogadott és elbírált diplomamunka elektronikus közzétételéhez (PDF formátumban a tanszéki honlapon) **hozzájárulok**.

Budapest, 2016. június 6.

a hallgató aláírása