



A projekt az Európai Unió támogatásával és az Európai Szociális Alap társfinanszírozásával valósul meg (támogatási szerződés száma TÁMOP 4.2.1/B-09/1/KMR-2010-0003)

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

WEBES INTERAKTÍV NÉVMUTATÓ KÉSZÍTÉSE VIRTUÁLIS GLÓBUSZOKON

SZAKDOLGOZAT
FÖLDTUDOMÁNYI ALAPSZAK

Készítette:

Tokai Tibor

térképész és geoinformatikus szakirányú hallgató

Témavezető:

Gede Mátyás

adjunktus

ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2011

Tartalomjegyzék

1. Bevezetés	3
2. Virtuális Glóbuszok Múzeuma	4
3. Perczel László és földgömbje	6
3.1. A gömb névanyagával foglalkozó tanulmányok	7
4. A webes interaktív névmutató	10
4.1. Alapanyagok és alkalmazott módszerek	10
4.2. HTML nyelv	12
4.3. A honlap elkészítése	13
4.3.1. Google Earth beépülő modul (Google Earth Plugin)	15
4.3.2. A névmutató elkészítése	19
4.3.2.1. Az keresési elemeket tartalmazó állomány létrehozása	19
4.3.2.2. MySQL adatbázis	22
4.3.2.3. Csatlakozás az adatbázishoz	23
4.3.3. Navigációs és kiegészítő eszközök	25
4.3.3.1. A navigációs eszközök elkészítése	25
4.3.3.2. A helyjelzők elkészítése	26
4.3.3.3. KML kódgeneráló gomb	28
4.3.3.4. Helyjelzők ki- és bekapcsolására szolgáló gomb	28
5. A honlap bemutatása	30
5.1. Nézőpont változtatás	31
5.2. A névmutató használata	33
5.3. A navigációs és kiegészítő eszközök használata	35
6. Összefoglalás	38
7. Köszönetnyilvánítás	39
8. Irodalomjegyzék	40
9. Nyilatkozat	43

1. Bevezetés

Napjainkban a kulturálisérték-mentés egyre fontosabb szerepet tölt be a tudományban. A technika fejlődése mellett nő a szakemberek felelőssége a különböző értékek megőrzésére, hiszen egyre tágabb a lehetőségek tárháza számukra e folyamatok végrehajtásához. Az értékmentés egyik módja napjainkban a digitális feldolgozás, restaurálás. Az ily módon történő digitalizálás segít megőrizni a szakmai és kulturális értékeket, és mindemellett felhívja a figyelmet ezeknek a fontosságára is. Térképészeti szempontból a régi glóbuszok feldolgozása is a kulturálisérték-mentés részét képezi, hiszen korukból és egyedülállóságukból adódóan nagy az eszmei értékük, viszont állapotuk az idők során erősen megromlott. Közülük kiemelkedő szerepet tölt be magyar, illetve talán világi viszonylatban is Perczel László földgömbje.

Véleményem szerint a műalkotás számos fontos sajátossága közül kiemelkedő szerepet tölt be a névanyaga. Korábban egyedülálló volt a részletessége. Ezért tartottam fontosnak, hogy a korábban végzett térképész hallgatók által kigyűjtött településneveket a gömbről valamilyen módon rendszerezem. Ehhez nem egy hagyományos névmutatót szándékoztam elkészíteni, mivel a technika segítségével már könnyebben használható megoldások is megvalósíthatóak. Emiatt határoztam el, hogy a tanszéken elkészült Virtuális Glóbuszok Múzeumának (a későbbiek során VGM) gömbjeihez hasonlóan elkészíték egy háromdimenziós glóbuszt Perczel László földgömbjéről, amelyet összekapcsolok a meglévő településnevekkel. Mindezt egy honlap formájában terveztem megvalósítani, hogy mindenki számára bármikor elérhető legyen. Szakdolgozatom célja ennek véghezvitele.

Munkám során bemutatom a VGM-et, Perczel Lászlót, valamint az általa elkészített glóbuszt. Ezek után részletesen leírom a webes interaktív névmutató elkészítésének technikai megvalósítását, azaz a honlap létrehozásához alkalmazott informatikai nyelveket, a gömb háromdimenziós megjelenítéséhez, illetve a nevekől létrehozott adatbázis kezeléséhez használt megoldásokat. Legvégül ismertetem az elkészült honlap funkcióit.

2. Virtuális Glóbuszok Múzeuma

A magyar föld- és éggömbkészítés történetéről, valamint magukról a glóbuszokról viszonylag csekély mennyiségű információt találunk a szakirodalomban. Mindössze egy komolyabb áttekintő írás született meg Klinghammer István tollából, „A föld- és éggömbök története”, amely azonban csak a múlt század első felével és az azt megelőző időszakokkal foglalkozik részletesebben. Fellelhetők még további írások az ezt követő időszakból, de ezek nem túl átfogóak.

Ezen hiányok pótlására jött létre 2008-ban a Virtuális Glóbuszok Múzeuma, az ELTE Térképtudományi és Geoinformatikai Tanszékén, amely honlap formájában lehetővé teszi ezeknek a műalkotásoknak a virtuális összegyűjtését, megőrzését és bemutatását. Azért is különösen fontos a VGM létrejötte, mert így az érdeklődőknek lehetőségük van egy időben és egy helyen szemügyre venni a magyar föld- és éggömbkészítés műalkotásait. Erre máskor sajnos nincs lehetőség, mert a valós életben különböző helyeken találhatóak meg.

A múzeumban a gömböket 3D-s formában tanulmányozhatjuk, forgathatjuk, tetszőleges távolságból szemlélhetjük. Sok esetben alaposabban vizsgálhatjuk őket, mint eredeti alakjukban. A 3D-s bemutatás mellett számos más információt is találhatunk róluk. A honlap rövid leírásokkal ismerteti a történetüket, a tartalmukat, az elkészítésükkor használt technológiát, valamint feltünteti a kiadó és a szerző nevét, kiadásuk helyét és idejét. Mindemellett sok esetben található fénykép is róluk, illetve glóbusztérképnyomat is, ha rendelkezésre áll. Természetesen nem mindig van lehetőség minden információ feltüntetésére a gömbökről, ugyanis sokszor hiányosak az ismeretek róluk. Ez különösen igaz a régebben készültrekre.

A gömbök digitalizálásának kétféle módja van. Az egyik lehetőség akkor alkalmazható, ha rendelkezésre áll térképnyomat a glóbuszról. Ebben az esetben a térképnyomat szkennelése után georeferálni kell a rajta lévő gömbkétszögeket, illetve pólussapkákat. Georeferálás után gömbre kell feszíteni a tartalmukat. Ennek az az előnye, hogy a térképnyomatok nem voltak kitéve az idő viszontagságainak, ugyanis elzárva, mappákban tárolták őket, tehát megőrizték a nyomtatásuk idejében meglévő élénk színüket.

Abban az esetben, ha nem áll rendelkezésre térképnyomat, akkor fényképsorozatot kell készíteni az eredeti gömbről. Ez az eljárás az alapja a digitális restaurálásnak is. A fotók elkészítésekor ügyelni kell arra, hogy a fényképezőgép optikai

tengelye átmenjen a gömb középpontján, valamint arra is, hogy a megvilágításból adódóan létrejövő becsillanás mértéke minél kisebb legyen. Természetesen nem minden részét lehet később felhasználni a fényképeknek, hanem csak egy előre meghatározott egységet. Ez köszönhető a már előbb említett becsillanásnak, a fénykép szélén jelentkező torzulásnak, illetve a fokhálózati csomópontok számának. A fotók száma a gömb méretétől függően változhat, a Perczel-gömb esetében kb. 800 fényképfelvétel készült. A meglévő felvételeket valamilyen térinformatikai szoftver segítségével lehet georeferálni. Erre az egyik legpraktikusabb program a Global Mapper. Könnyű kezelhetőségének köszönhetően igen gyorsan és hatékonyan lehet vele dolgozni. A georeferálás úgy történik, hogy a programban megnyitjuk az elkészült fényképeket, majd illesztőpontokat, úgynevezett GCP-eket (Ground Control Point) hozunk létre a fokhálózati metszéspontok megjelölésével. Ezeknek a pontoknak meg kell adni a koordinátáját, amit egyszerűen meg lehet állapítani a fokhálózat segítségével. Végül különböző vetületi transzformációk elvégzése után megkapjuk a kész, georeferált állományunkat. (MÁRTON 2009)

A virtuális múzeum 3D-s elemeinek a megjelenítését a VRML nyelv használata tette lehetővé. A VRML egy egyszerű scriptnyelv, amellyel háromdimenziós világokat lehet létrehozni a világhálón. Elkészítéséhez csupán egy egyszerű szövegszerkesztő program szükséges, az elkészült fájl kiterjesztésének .wrl-nek vagy .wrz-nek kell lennie. A létrehozott világot felépítő objektumokat, azok geometriáját, kinézetét és egymáshoz való viszonyát bizonyos szintaktikai szabályok szerint kell leírni. Az előzőekben említett kiterjesztések valamelyikével elkészített fájlt a böngészők értelmezik, majd megjelenítik a készítő által megálmodott világot. Az így létrehozott fájlok megtekintéséhez egy VRML lejátszó program telepítése szükséges (pl. Flux Player, Cortona3D).

A 3D-s megjelenítés másik módja a Google Earth használata. Ebben az esetben a georeferált térképtartalomból készített KML fájl egy réteggént viselkedik a programban, amellyel különböző műveleteket hajthatunk végre.

A megannyi gömb közül, amit a Virtuális Glóbuszok Múzeumában megtalálhatunk, úgy gondolom, hogy magyar viszonylatban az egyik legértékesebb alkotás a Perczel-féle gömb.

3. Perczel László és földgömbje

A föld- és éggömbkészítésben már a kezdetektől fogva fontos szerepet töltenek be magyar szakemberek. Közéjük tartozik a 19. század egyik kiváló alkotását készítő Perczel László is.

Perczel László Börzsönyben született 1827-ben. Élete korai éveiben katonai tevékenységet folytatott. 1841-től a bécsújhelyi Katonai Akadémián tanult, amelyet 1846-ban végzett el. Ezután hadnaggyá avatták és tisztként szolgált a varasdi katonai határőrvidéki ezrednél. 1848–49-ben részt vett az itáliai szabadságharc elfojtásában. A szabadságharc után négyévi várfogságra ítélték, azonban idő előtt szabadulhatott, és 1850-ben saját kérésére elbocsátották a hadseregből. Két évvel később megnősült és kömlődi birtokára költözött. Itt kezdte el készíteni a 19. század egyik legjelesebb magyar vonatkozású műalkotását, a Perczel-glóbuszt, amely 1862-re készült el és az elkészítése egyes források szerint legalább kilenc évet vehetett igénybe. Arról, hogy milyen forrásokból dolgozott, valamint, hogy volt-e segítőtársa, nem tudunk.



1. ábra - Perczel László földgömbje

A Perczel-gömb egy kézirat, 132 cm átmérőjű, 10 milliós méretarányú óriás földgömb. Részletesebb leírást Ambrus-Fallenbüchl Zoltán 1963-as feljegyzéseiben találhatunk róla, amely azért is nagyon fontos, mert az 1970-es évekbeli restaurálás előtt született. Az akkor elvégzett restaurálás során sajnos többet ártottak a gömbnek, mint amennyi haszon lett volna belőle. A nem megfelelő restaurálási technikák alkalmazása miatt, a gömbön levő lakkréteg mára erősen besárgult, s emiatt egyes megírások már nehezen vagy egyáltalán nem olvashatóak. Ez különösen igaz az egykor vörös színnel megírt településekre, illetve az őket jelző pontokra. Ezek a problémák mellett még a gömbön, illetve a tartóállványon levő további sérülések is nehezítik vagy lehetetlenné teszik a gömb egyes részeinek a vizsgálatát.

Perczel László a gömb készítésekor a rengeteg településnév mellett ábrázolt vízrajzot, színes határvonalakat, árnyékolt hegyrajzot, valamint feltüntette a tengeráramlásokat is. Elkészültekor egyedülálló volt pontossága és részletes megírása miatt. Becslések szerint a gömb legkevesebb 15–18 000 földrajzi nevet tartalmazhat. A részletességére utal, hogy csak a mai Magyarország területén tizenegy vízrajzi név, és tizennégy településnév található. Ehhez még azt is fontos hozzátennem, hogy annak ellenére, hogy a gömb magyar alkotó által jött létre, nem ábrázolja Magyarországot részletesebben többi területnél. Igyekezett a településneveket mindig magyar nyelven írni, de nem minden esetben létezett magyar nevük. A földrajzi közneveket mindig magyar nyelven írta meg. A következőkben a gömb névanyagával foglalkozok részletesebben. (MÁRTON 2008)

3.1. A gömb névanyagával foglalkozó tanulmányok

A gömb névanyagával részletesebben Ambrus-Fallenbüchl foglalkozott a már említett munkájában. Részletes leírást ad a gömb akkori állapotáról, a vázától kezdve, a tartalmán át, egészen az állványzatáig. Engem elsősorban a glóbusz tartalmi erényei érintenek, azon belül is annak névanyaga. Mintaszerű munkájában aprólékosan leírja, hogy Perczel László mennyire egyedülállóan részletes névanyagú gömböt készített el a maga korában. Foglalkozik a különféle földrajzi objektumok neveivel, megírásuk módjával, színhasználatával, és a színek ellenállóságával.

„A településeket vörös pontok jelzik : névírásuk is vörös. A folyók névírása kék, a hegységeké barna: ez utóbbi megjelölések ritkábban, mintsem várni lehetne.

Sajnálatos módon a települések vörös névírása az idők folyamán meglehetősen elfakult, helyenként teljesen olvashatatlaná vált. Jóval ellenállóbbnak bizonyult a folyók kék névírása.”

(AMBRUS-FALLENBÜCHL 1963)

Mélyrehatóan elemzi a települések neveit, utalásokat tesz azok mennyiségére, előfordulására példákon keresztül. Magyarország példáján keresztül leírja, hogy a sűrűbben lakott területek névanyaga mennyire részletes.

„Hogy a sűrűbben lakott területek névírása milyen részletes, arra példaként éppen Magyarország szolgálhat, melyet a készítő távolról sem részesített kivételes elbánásban. A mai magyar államterület néhány négyzetcentiméterén a Dunán, Dráván, Tiszán kívül a Rába, Zala, Sárvíz, Zagyva, Körös és Maros folyók, a Fertő és a Balaton tavak is rajta vannak : a városok közül Buda, Pest, Fehérvár, Győr, Veszprém, Pécs, Vác, Eger, Kecskemét, Szeged, Csongrád, Gyula, Debreczen, Sopron az elmosódott írás miatt alig található meg.”

(AMBRUS-FALLENBÜCHL 1963)

Arra következtetett, hogy Perczel László teljes mértékben tisztában volt a korában megtörtént legújabb földrajzi felfedezésekkel, ugyanis gömbjén olyan neveket is feltüntetett, amelyeket még az európai földrajztudósok sem ismertek. Megállapítja, hogy azon nevek, amelyeknek van magyar alakjuk is, a gömbön is úgy szerepelnek. Tehát az alkotó törekedett minél nagyobb részben a névanyag tartalmát magyar alakban megjeleníteni. (AMBRUS-FALLENBÜCHL 1963)

A Perczel-glóbuszhoz készített webes névmutató megvalósításához három korábban végzett térképész hallgató diplomamunkájához létrehozott geoadatbázisokat használtam fel. Nyuli Éva foglalkozott Európa településneveivel, Tóth Bettina Dél-Amerikáéval, Való Adrienn pedig Afrika településeinek a névanyagával. Munkájuk során a névanyag megőrzéséhez és kigyűjtéséhez dolgoztak ki különféle praktikákat. A glóbusz megőrzésében nagyon fontos lépésnek tartom ezt, tekintve a névanyag mai állapotát. Az általuk kigyűjtött nevek nagy segítségemre voltak a webes névmutató elkészítésében.

Mivel a gömbről nem állt rendelkezésre térképanyomat, ezért fényképsorozatot készítettek róla. A fotókat elosztották, és térinformatikai szoftver (Global Mapper)

segítségével georeferálták őket. Próbálták minél pontosabban leolvasni az egyes kontinensek településneveit, majd a program segítségével megállapították a települések koordinátáit. Minden településnévnek kétféle változatát gyűjtötték ki az adatbázisba, ha volt rá lehetőség. Az első oszlopba helyezték az általuk leolvasott neveket. A nevekben sok esetben kérdőjel is szerepel, ez arra utal, hogy a gömbön szereplő név, vagy annak valamelyik karaktere, teljes mértékben olvashatatlan volt. Emiatt hoztak létre egy második oszlopot is, ahova a nevek feltételezett alakját írták be. Ehhez sok esetben korabeli térképek segítségét vették igénybe, ám sajnos még így sem tudtak minden névhez valamilyen valószínűsített nevet írni. Annak érdekében, hogy minél pontosabb legyen a munkájuk létrehoztak egy harmadik oszlopot is, amelyben egy igaz/hamis (T vagy F) logikai érték szerepel, ami arra utal, hogy az adott településnév olvasható volt-e vagy sem. (NYULI; TÓTH; VALÓ 2009)

A következő fejezetben bemutatom a webes névmutató megvalósításának az okát és elkészítésének a részleteit.

4. A webes interaktív névmutató

A webes névmutató megvalósításának az ötlete több dolog miatt is felmerült. Az egyik ok, a korábban kigyűjtött településnevek valamilyen formában való összegzésének a szüksége volt. Úgy gondolom, hogy egy ilyen átfogó munka megalkotásához a legjobb módszer egy névmutató elkészítése, amely összeköti a neveket a térkép hozzájuk tartozó részeivel. Egy ilyen webes névmutató működésében különbözik a hagyományostól, itt oldalszámok, valamint betűk és számjelzések helyett egy függvény segít a névanyag és a térkép összekapcsolásában. Használata jóval egyszerűbb a megszokottnál, hiszen egy adott településnév keresésénél nem kell lapozgatni, illetve a sorok között keresgélni, mint például egy atlaszban, hanem csak be kell írni a nevet egy keresőbe, ezután a böngésző rögtön kiválasztja a keresett települést egy listából, majd meg is tudja mutatni annak helyét a gömbön 3D-s formában.

A másik ok az volt, hogy a jövő évben, 2012-ben ünnepli a gömb a 150. évfordulóját, így időszerű volt több időt fordítani a megőrzésének fejlesztésére. Amellett, hogy tervben áll a fent jelzett időpontig többek között a glóbusz virtuális restaurálásának, illetve egy digitális faksimile készítésének a megvalósítása, véleményem szerint nagy értéket képviselne a virtuális névmutató teljes elkészülése is. Ehhez persze szükség lenne az eddig kigyűjtött három kontinens névanyaga mellett a többi kontinens neveinek a kigyűjtésére is. (MÁRTON 2008)

4.1. Alapanyagok és alkalmazott módszerek

A munka elkészítéséhez alapanyagként felhasználtam a már meglévő adatbázisokat a névanyagról, valamint a gömb georeferált fotóiból készült KML fájlt. A KML (Keyhole Markup Language) egy XML-alapú jelölőnyelv, amely térben ábrázolt alakzatok megjelenítésére szolgál a Google térképészeti vonatkozású alkalmazásaiban. Ide sorolható a Google Earth szoftver, illetve a Google Maps számítógépes és mobiltelefonos változata. A fájl bizonyos tulajdonságokat határozhat meg, amelyek vonatkozhatnak például a kamera nézetére, a textúrára, vagy a földrajzi helyzetre.

A három adatbázis adatainak egységesítéséhez a Microsoft Excel táblázatkezelő szoftvert használtam. A program segítségével összegeztem egy új munkalapon a kontinensekről kigyűjtött adatokat, majd rendszereztem őket. Külön oszlopba

rendeztem a szélességeket, a hosszúságokat, a leolvasott neveket és a feltételezett település neveket.

	A	B	C	D
1	LONGITUDE (G)	LATITUDE (G)	leolvasott_név	valószínű_név
2	16.313855	48.050572	Bécs	Bécs
3	16.440391	47.792633	U????	?
4	16.655038	47.542221	Sopron	Sopron
5	14.624191	46.050572	Laibach	Laibach
6	14.506119	47.955474	Stey?	Steyr
7	15.411341	47.168324	Gratz	Gratz
8	14.128287	48.175876	Linz	Linz
9	14.521862	48.703267	Budweis	Budweis
10	15.521542	49.443188	Jglau	Iglau
11	14.805236	49.372344	Tubor	Tabor
12	17.072227	49.758048	Olm?t?	Olmütz
13	16.568451	49.356601	K?	Kremsier
14	17.08797	48.199491	Po?	Pozsony
15	18.874801	47.640614	Budapest	Budapest
16	18.992873	47.853145	Vác	Vác
17	18.048293	47.805916	Ko??om	Komárom
18	17.528774	47.727201	Gy?	Győr
19	18.780343	48.396278	Beszterce?	Besztercebánya
20	18.030403	48.91365	T??csén	Trencsén
21	18.662986	48.67035	???	???
22	20.311349	47.934364	Eger	Eger
23	21.540019	47.551166	Debrecen	Debrecen
24	21.029087	48.658184	K??	Kassa
25	21.150737	48.88932	??	??
26	23.705396	49.935514	Lambarg	Lambarg
27	22.835596	49.546233	S??bor	??
28	25.213862	49.710461	Tur??	Tarnapol
29	24.636022	48.579112	Kolom??	Kolomea

2. ábra – Részlet az Excel táblában összegzett adatokról

1862-ben, a gömb elkészültekor, még a ferrói kezdőmeridián volt az alapértelmezett. Emiatt át kellett számítanom a hosszúsági értékeket, az 1884-től kezdődően, ma is használatban levő greenwichi kezdőmeridiánra. Kivontam minden egyes hosszúsági értékből 17,66277°-ot (kb. 17°40'), ugyanis ez az érték adja meg Ferro és Greenwich hosszúságkülönbségét. Erre azért volt szükség, mert a virtuális névmutató elkészítéséhez a gömb 3D-s modelljét a Google Earth beépülő moduljának segítségével készítettem el, ami ugyancsak greenwichi kezdőmeridiánt alkalmaz. (TÍMÁR 2007)

Ahhoz, hogy a beépülő modul háromdimenziós gömbje ne az alapértelmezett műholdképekből összeállított képet mutassa, hanem a Perczel László-féle gömbét, egy a róla készült georeferált fotókból létrehozott KML fájlra volt szükségem. Ez a fájl jelen esetben egyfajta textúráként szolgál, segítségével nem csak egy egyszerű földgömböt láthatunk, hanem a Perczel-glóbusz eredeti arcát. Létrehozása már korábban megtörtént a VGM keretein belül, csupán minimális felbontásbeli különbség van az ott használt, és az általam alkalmazott fájl képei között. Erre azért volt szükség, mert a VGM-ben

alkalmazott fájl létrehozásához használt fényképek nem voltak elég nagy felbontásúak a névanyag gömbön való vizsgálásához.

A honlap kódjának a megírásához a Notepad++ szöveg- és forráskódszerkesztő programot használtam, amely rendkívül praktikus erre a célra szintaxis kiemelésé, illetve egyéb más funkciói miatt. A hozzá tartozó CSS fájlt az Adobe Dreamweaver CS4 programmal írtam meg és hoztam létre.

4.2. HTML nyelv

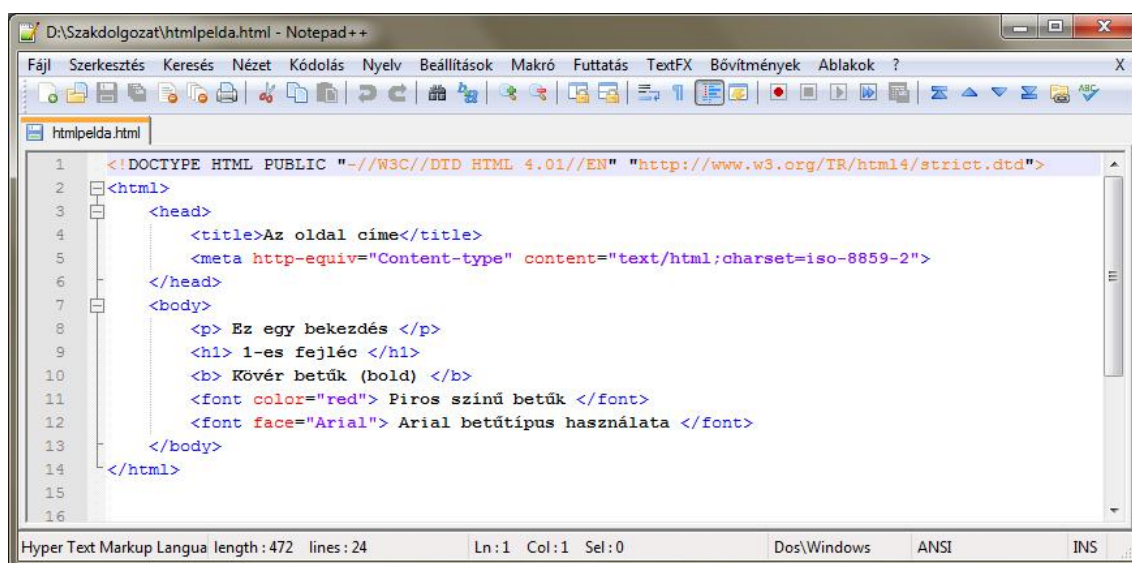
A honlap megírásához a HTML leíró nyelvet használtam, ami bármilyen szövegszerkesztő program segítségével létrehozható. A HTML egy angol mozaikszó, jelentése HyperText Markup Language (hiperszöveges jelölőnyelv), amit a World Wide Web Consortium fejlesztett ki honlapok készítéséhez és szerkesztéséhez. A legelterjedtebb változata 4.01-es, amit 1999-ben adtak ki és azóta is széles körben alkalmazzák.

Egy HTML dokumentum alapvetően HTML-elemekből, más néven címkékből (tagekből) épül fel. Ezek az elemek magukba foglalhatnak különféle jellemzőket (attribútumokat) és tartalmakat. Jelölhetnek bekezdéseket, betűtípusokat, stílusokat, fejléceket, hipervivatkozásokat és sok egyéb más jellemzőt is. Egy HTML-elem a következő fő részekből tevődik össze:

- Kezdő címke, ami az elem kezdetét jelöli
- Attribútum, ami a kezdőcímkén belül helyezkedik el és bizonyos jellemzőket és a hozzájuk tartozó értékeket jelöli
- Tartalom, ami a kezdő és záró címke között helyezkedik el és bármilyen tartalmat tartalmazhat (szöveg, beágyazott elemek stb.)
- Záró címke, ami az elem végét jelöli

Egy HTML dokumentum három fő részből áll és mindig <HTML> utasítással kezdődik és </HTML> záró utasítással végződik. Az első rész a dokumentumtípus definíció (DTD), amivel megadhatjuk, hogy a HTML melyik verzióját akarjuk használni, illetve még segítségünkre szolgál abban is, hogy az elkészített honlapunk hasonló képet mutasson a különböző típusú és gyártmányú böngészőkben. Ez a szegmens a <HTML> elem előtt helyezkedik el. A másik fő rész a fejléc, amely technikai és dokumentációs adatokat tartalmaz és a <HEAD>...</HEAD> utasítások között helyezkedik el. Ezt a

részét a honlapunknak a böngésző nem jeleníti meg. A harmadik rész a honlap törzse, amely a honlap valódi, közölni kívánt tartalmát foglalja magában és a <BODY>...</BODY> elemek között található.

The image shows a Notepad++ window titled 'D:\Szakdolgozat\htmlpelda.html - Notepad++'. The menu bar includes 'Fájl', 'Szerkesztés', 'Keresés', 'Nézet', 'Kódolás', 'Nyelv', 'Beállítások', 'Makró', 'Futtatás', 'TextFX', 'Bővítmények', 'Ablakok', and '?'. The toolbar contains various icons for file operations and editing. The main text area shows the following HTML code:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html>
3 <head>
4     <title>Az oldal címe</title>
5     <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2">
6 </head>
7 <body>
8     <p> Ez egy bekezdés </p>
9     <h1> 1-es fejléc </h1>
10    <b> Kővér betűk (bold) </b>
11    <font color="red"> Piros színű betűk </font>
12    <font face="Arial"> Arial betűtípus használata </font>
13 </body>
14 </html>
15
16
```

The status bar at the bottom indicates 'Hyper Text Markup Langua length : 472 lines : 24', 'Ln:1 Col:1 Sel:0', 'Dos\Windows', 'ANSI', and 'INS'.

3. ábra - Egyszerű példa egy HTML dokumentumra a Notepad++ programban

4.3. A honlap elkészítése

Elsőként a honlap szerkezetének az elkészítését mutatom be, amelyhez a HTML nyelvben széleskörűen alkalmazott <DIV> elemeket használtam. Ezeket az elemeket a blokk szintű tartalmazó elemekhez sorolják, mégpedig azért, mert mindig magukba foglalnak valamilyen másik tartalmat, akár egy újabb <DIV> elemet is. Tulajdonképpen oly módon működnek mintha dobozok lennének, amelyeket egymásba helyezünk. Lehetővé teszik a honlap szerkesztőjének, hogy a <DIV>...</DIV> elemek között levő tartalom egészére írjon elő egy stílust vagy valami más formázási műveletet.

A <DIV> elemekhez egyszerűen rendelhetünk stílusokat. Mindössze annyit kell tennünk, hogy a parancs attribútumaként megadunk egy azonosítót névvel ellátva (pl. id="valami"), amely alapján később a stíluslapban majd hivatkozunk rá. A stílusok meghatározását a CSS használatával van lehetőségünk elvégezni, aminek jelentése Cascading Style Sheets, vagyis egymásba ágyazott stíluslapok. Ez egy egyszerű stílusleíró nyelv, amely nagymértékben megkönnyíti az oldalunk formázását. Segítségével megváltoztathatjuk a HTML elemek színét, méretét, elhelyezkedését, vagy egyéb más tulajdonságát. Természetesen nem csak <DIV> elemekre alkalmazható, hanem például bekezdésekre, címsorokra is.

A stíluslapokat többféleképpen is hozzákapcsolhatjuk a HTML állományunkhoz, de leginkább két mód alkalmazása terjedt el. Az egyik lehetőség, hogy a stílusleírást a HTML oldal fejlécében hozzuk létre, `<style type="text/css">...</style>` elemek közé zárva. Ezt az eljárást beágyazott stíluslapnak hívjuk. A másik lehetőség, hogy készítünk egy külső stíluslapot. Ilyenkor a stíluslapunkat elhelyezzük egy külső állományban, amelyet `.css` kiterjesztéssel elmentünk, majd az oldal fejlécében hivatkozunk rá egy `<LINK>` elem segítségével. Ennek az az előnye, hogy a létrehozott fájlra akár több HTML dokumentumból is hivatkozhatunk, illetve az állományunk is átláthatóbb lesz, mivel a CSS elemek átkerülnek belőle egy külön fájlba.

A honlap elkészítéséhez az utóbbi eljárást alkalmaztam. Elkészítettem egy külső stíluslapot, amelyet *stilus.css* néven neveztem el, majd ezt hívtam meg az *index.html* nevű állományomból. A CSS fájlomban szinte kivétel nélkül csak `<DIV>` elemekre hivatkoztam. Ezek az elemek, illetve a stílusleírások segítségével építettem fel a honlap szerkezetét.

Alapvetően a honlap lényegi része egy *content* elnevezésű azonosítóval ellátott `<DIV>` elemen belül helyezkedik el. Ebben találhatóak a címet tartalmazó, a Google Earth beépülő modul gömbjét, a névmutatót, illetve a navigációs eszközöket tartalmazó `<DIV>` elemek. Mindegyiknek megadtam egy azonosítót (id), amelynek segítségével tudtam formázni őket a CSS állományomban az igényeim szerint. Ily módon állítottam be minden egyes ilyen elemnek a méretét, a színét, a belsejükben használt margók méretét, és az elemek közötti távolságok nagyságát (padding, vagyis kitöltés).

A honlap tartalmi részének a méretét igyekeztem úgy meghatározni, hogy bármilyen esetben, azaz akármilyen monitorral vagy felbontással kényelmesen lehessen használni, illetve, hogy mindig esztétikus képet mutasson. Ehhez mérten szabtam meg az egyes `<DIV>` elemek méretét is. Színezésükben próbáltam megtalálni a harmóniát, figyelembe véve a Perczel-gömb világosbarna színét is. Ennek következtében az egész tartalmi részt magába foglaló *content* azonosítójú elemnek egy sötétbarna színű árnyalatot választottam, és az ezen belül található `<DIV>` elemeknek egy, a gömbnél sötétebb árnyalatú, de az előbb említetténél világosabb, barna színt alkalmaztam. Az egész honlap háttérének beállítottam egy viszonylag semleges színárnyalatú világosszürke színt. Ezt is a CSS állományban határoztam meg. A látványhoz hozzátartozik még az is, hogy az összes ilyen elemnek adtam egy vékony szegélyt is a háttérrel megegyező színben. Ez alól kivétel a *content* azonosítójú elem, ugyanis annak a színe a cím színével egyezik meg. Mindezek mellett beállítottam még a margókat,

valamint a <DIV> elemek közötti távolságokat is.

Eredetileg a honlap mindegyik létrehozott és szegéllyel ellátott <DIV> elemet hegyes sarkú négyszöggként jelenít meg. Ennek kiküszöbölésére, illetve a honlap küllemének a javítására lekerekítettem ezeknek a „dobozkáknak” a sarkait. Ezt egy *border-radius* nevű parancs használatával, és a hozzátartozó értékek megadásával tettem meg a CSS állományban. Sajnos ez a funkció Internet Explorer használatával nem működik, úgyhogy a honlap böngészéséhez Mozilla Firefox használatát ajánlom.

Mivel az első elem csak a honlap címét tartalmazza, ezért nem tértem külön ki rá. A következő részben a második <DIV> elemről, pontosabban annak tartalmáról, a Google Earth beépülő modulról fogok részletesebb leírást adni.

4.3.1. Google Earth beépülő modul (Google Earth Plugin)

A Google Earth beépülő modul, hasonlóan az önálló változathoz, lehetővé teszi a Föld 3D nézetben való szemlélését az interneten keresztül, anélkül, hogy telepítenünk kellene a teljes Google Earth szoftvert. Alkalmazásprogramozási felületének (API – Application Programming Interface) segítségével számos, az eredeti szoftverből már ismert funkcióját élvezhetjük az alkalmazásnak. Segítségével létrehozhatunk helyjelzőket és vonalakat, képeket, illetve 3D modelleket (pl. épületeket) helyezhetünk el a felszínen, és betölthetünk KML fájlokat is, amelyek további alternatívákat tesznek lehetővé. A Google Earth Plugin által Windowsos környezetben támogatott böngészők a következők:

- Google Chrome 1.0+
- Internet Explorer 6.0+
- Firefox 2.0+, Flock 1.0+

Telepítése és használata akkor is lehetséges, ha nincsen a teljes Google Earth szoftver installálva a számítógépünkre. Ekkor csupán annyit kell tennünk, hogy meglátogatunk egy olyan honlapot, amelyen megtalálható ez az alkalmazás, majd a böngészőnk automatikusan felajánlja számunkra a modul telepítését.

Ahhoz, hogy a modult a saját honlapomon tudjam alkalmazni, szükségem volt néhány dolog elvégzésére. Először betöltöttem a Google Earth API-t, elkészítettem egy <DIV> tároló elemet a modulhoz, majd létrehoztam bizonyos függvényeket a plugin inicializálásához.

A Google Earth API betöltéséhez szükségem volt egy igen hosszú, számokból és betűkből álló API kulcsra, amelyért ingyenesen lehet jelentkezni az alkalmazás honlapján. A birtokomban levő kulcsot beillesztettem a honlap fejlécébe egy <SCRIPT> elembe ágyazva a következő módon:

```
<script type="text/javascript" src="http://www.google.com/jsapi?key= API kulcs helye ">
```

Az elem *src* része egy JavaScript fájlra mutat egy egyszerű metódus segítségével, a *google.load*-dal.

```
google.load("earth", "1");
```

A JavaScript egy széles körben elterjedt, webes programozáshoz használt, objektumalapú szkriptnyelv. A későbbiek során számos ilyen nyelven megírt függvényt fogok alkalmazni. (MONCUR 2002)

Az előbb említett metódust egy új <SCRIPT> elembe helyeztem el. Használata lehetőséget ad arra is, hogy egyidejűleg több különböző Google API legyen betöltve a honlapra. A zárójelen belüli paraméterek arra szolgálnak, hogy utasítsák a Google-t az Earth modul első verziójának a betöltésére, ami az API utolsó stabil verziója.

A beépülő modul honlapon való elhelyezését egy <DIV> tároló elem tette lehetővé. Miután ebbe az elembe betöltöttem a modult, elhelyeztem azt a honlap törzsébe. Ezek után adtam neki egy azonosítót, amelynek közreműködésével hozzá tudtam rendelni a modult az elemhez. Az azonosítót *map3d* néven neveztem el. Az elembe írt *style* attribútummal változtatni lehet a modul bizonyos tulajdonságain. Segítségével meghatároztam, hogy mekkora legyen a háromdimenziós gömböt tartalmazó ablak szélessége és magassága (*width*, *height*), illetve a *border* paranccsal hozzárendeltem egy szegélyt is, amit egy pixel szélességűre állítottam.

```
<div id='map3d' style='border: 1px solid silver; height: 600px; width: 800px;'></div>
```

A modul kezdeti működésének előkészítéséhez, a következő alapvető függvényekre volt szükségem:

```
function init() {
    google.earth.createInstance("map3d", initCB, failureCB);
}

function initCB(instance) {
    ge = instance;
}
```

```

    ge.getWindow().setVisibility(true);

    // no atmosphere
    ge.getOptions().setAtmosphereVisibility(false);

    // no terrain
    ge.getLayerRoot().enableLayerById(ge.LAYER_TERRAIN, false);

    // fetch the KML
    var url='http://mercator.elte.hu/~totqaat/perczel_006.kmz';
    google.earth.fetchKml(ge, url, finished);
}

function failureCB(errorCode) {
}

```

Az *init* elnevezésű függvény lényege, hogy megpróbálja létrehozni a modult. Alapvetően három opciója van. Az első helyére az előzőekben létrehozott <DIV> elemnek az azonosítóját írtam be, mert ahhoz szerettem volna hozzárendelni a modult. A második opcióhoz annak a függvénynek a nevét adtam meg, amelyet akkor hív meg az *init* függvény, ha a modul betöltése sikeres volt. Ezt a függvényt *initCB*-nek nevezik. A harmadik opció helyére a *failureCB* elnevezésű függvény nevét írtam be, ez (ahogy a neve is mutatja), akkor fut le, ha valamiért a modul nem tud sikeresen létrejönni.

Az *initCB* elnevezésű függvény, azokat az információkat tartalmazza, amelyeknek a modul első lefutásánál létre kell jönniük. A függvénynek mindenképp tartalmaznia kell a *GEWindow.setVisibility* metódust *true* értékkel megadva, mert csak ennek a segítségével tud megjeleníteni a modul a <DIV> elemen belül. A függvényben meg lehet adni további parancsokat is. Jelen esetben kikapcsoltam a Google Earth földgömb atmoszférájának a megjelenítését, mert szükségtelen volt a későbbieknek, illetve a terep részletes megjelenítését is megszüntettem. Ez annyit jelent, hogy a hozzájuk tartozó parancsok *false* értéket adtam meg. Mindezek mellett, ebben a függvényben gondoskodtam arról is, hogy a Google ne az eredetileg látható, műholdképekből összeállított gömb képét mutassa, hanem a Perczel-glóbusz eredeti alakját. Erre a *google.earth.fetchKml* parancs segítségével volt lehetőségem, amelyben a korábbiakban elkészített, georeferált fényképekből összeállított KMZ fájl elérési útját adtam meg, és így módon a függvény betöltötte a fájlt a beépülő modulba. Hasonló eljárással töltöttem be az egyes településekhez tartozó helyjelzőket is, erről majd egy későbbi fejezetben részletesebben tárgyalok.

A *failureCB* nevű függvény abban az esetben jön létre, ha a modul nem fut le sikeresen. Alapvetően a belseje tartalmazhatna bizonyos kódokat, amelyek arra szolgálhatnának, hogy például kiírjanak egy hibaüzenetet a képernyőre, ezt nem tartottam szükségesnek, ezért üresen hagytam azt.

Az alapvető függvények mellett további függvényeket is alkalmaztam a honlap megfelelő működéséhez:

```
function setVP(fila,range) {
    vpt=fila.split(',')
    var vp=ge.createLookAt('');
    vp.set(vpt[0]*1, vpt[1]*1, range,
    ge.ALTITUDE_RELATIVE_TO_GROUND, 0, 0, 0);
    ge.getView().setAbstractView(vp);
}

function finished(object) {
    if (!object) {
        // wrap alerts in API callbacks and event handlers
        // in a setTimeout to prevent deadlock in some browsers
        setTimeout(function() {
            alert('Bad or null KML.');
```

A *setVP* függvény arra szolgál, hogy a megadott paramétereknek megfelelően beállítsa a gömb állását a kívánt pozícióba. A függvény paramétereiként egy földrajzi koordinátapárt, valamint a kívánt szemmagasságot értem méterben meghatározva. Megadásukkor, a függvény meghívása után, idézőjelek közé kell beírni azt a koordinátapárt, ahová be szeretnénk állítani a nézetet, mint ahogy az a következő, *finished* elnevezésű függvény legvégén is látható. A koordinátapárt minden esetben vesszővel kell elválasztani, mert a függvény így képes elkülöníteni a két koordinátát. Mellettük egy újabb vesszővel elválasztva kell megadni a kívánt szemmagasságot is. Esetemben a praktikus megoldás az volt a koordinátapár megadásával kapcsolatban, hogy egy külső forrásból rendeltem a függvényhez a megfelelő értékeket. Ez a külső forrás egy MySQL adatbázis volt.

A *finished* függvény megakadályozza egyes böngészők összeomlását a modul használatakor. Egy *setTimeout* parancsal egy bizonyos idő elteltével meghív egy újabb függvényt, amely egy hibaüzenettel tér vissza. Ez az időtartam milliszekundum egységként értendő. Ha ez a hiba nem következik be, akkor beállítja a földgömb nézetét az előbbieken leírtak szerint a keleti hosszúság 20°-ára és az északi szélesség 50°-ára.

A következő feladatom a névmutató elkészítése és a hozzá tartozó keresési funkciók összeállítása volt.

4.3.2. A névmutató elkészítése

A névmutató alapvetően különböző keresési elemekből tevődik össze, amelyeknek az oldal elkészültekor az lesz a funkciójuk, hogy az általunk keresett település koordinátáját elküldik annak a függvénynek, amely beállítja a Google Earth földgömböt a megfelelő pozícióba. Ez annyit tesz, hogy a gömböt úgy forgatja el, hogy a keresett település legyen középpontban a modul megjelenítendő részén.

Mindegyik kereséshez szükséges elemet egy külön <DIV> elembe helyeztem el. Közéjük tartozik egy szövegdoboz, egy keresést feldolgozó gomb, és egy lista, amely tartalmazza az összes település nevét. A szövegdoboznak és a listának a működése össze van hangolva. Ha a szövegdobozba beírjuk egy keresni kívánt település nevét, akkor a lista automatikusan ahhoz a településhez ugrik. Ez minden egyes karakter begépelésénél folyamatosan változik, mígnem a begépelte szöveg el nem éri a végleges alakját. Ezeket az elemeket egy külön PHP fájlban hoztam létre, amelyet *kereso.php* néven neveztem el. Erre azért volt szükségem, mert az összegyűjtött adatokat egy MySQL adatbázisban összegeztem, és ebből az adatbázisból csak a PHP nyelv használatával tudtam kinyerni őket.

Miután elkészült ez az állomány, a honlap fő részéből (*index.html*) egy *kereso* nevű azonosítóval ellátott <DIV> elemen belül elhelyezett *iframe* HTML elem segítségével hívtam meg. Ez az elem lehetővé teszi dokumentumok beágyazását a fő HTML dokumentumba. Méretét a honlap fő részében lehet meghatározni, ott ahol a meghíváshoz használtam. Úgy állítottam be ezt a tulajdonságot, hogy a meghívott fájlban található keresési elemek megfelelően elférjenek a számára létrehozott <DIV> elemekben, és ne lógjanak túl rajta.

4.3.2.1. Az keresési elemeket tartalmazó állomány létrehozása

A keresési elemeket tartalmazó PHP fájl létrehozásához szükségem volt egy JavaScript függvény használatára, amely elősegíti a kereséshez használt szövegdoboz és a lista összehangolását. Továbbá használnom kellett azokat a HTML kódokat, amelyek segítségével létrehozhattam a keresési elemeket. Végül az összegyűjtött településneveket és a hozzájuk tartozó adatokat egy MySQL adatbázisban összegeztem, majd a PHP nyelv használatával feltöltöttem a keresési listát velük.

Az összehangolást segítő JavaScript függvény a következőképpen néz ki:

```
function findoption(strText, strOption) {
    objForm = document.forms[0];
    objSelect = objForm.elements[strOption];
    intStringLength = strText.length;
    for(i=0; i<objSelect.length; i++) {
        strDisplayText =
objSelect.options[i].text.substring(0, intStringLength);
        if(strDisplayText.toUpperCase() == strText.toUpperCase()) {
            objSelect.selectedIndex=i;
            break;
        }
    }
}
```

A függvényt *findoption* névvel láttam el, és az a feladata, hogy ha a felhasználó egymás után begépel karaktereket, akkor megvizsgálja azt, hogy az általunk figyelni kívánt elemek között van-e a begépeléssel megegyező karakterekből álló elem. Esetemben ezek az elemek a településnevekkel egyeznek meg. Ha a begépeltektársorozatok nem, vagy nem teljesen azonosak a meglévő településnevek egyikével sem, a függvény akkor is megkeresi a karaktereiben leginkább hasonlító nevet. Ha létezik a településnevek között a begépeléssel teljesen megegyező karakterekből álló név, akkor természetesen azt is kiválasztja. Mindezt egy FOR ciklus keretein belül végzi el. A függvénynek előnye, hogy begépelésnél nem különbözteti meg a kis- és nagybetűs karaktereket.

A kereséshez használt elemeket és az őket tároló táblázatot HTML kódok segítségével hoztam létre. A szövegdozst, amelybe a keresni kívánt település nevét kell beírni, valamint a keresést feldolgozó gombot egyaránt egy *input* paranccsal valósítottam meg.

A szövegdozst elkészítéséhez egy szöveg típusú *input* elemet kellett használnom:

```
<input type="text" name="minta" onkeyup=findoption(this.value, 'list')>
```

Az *onkeyup* paraméter értékeként a JavaScript függvény nevét írtam be, ennek köszönhetően lép működésbe gépeléskor a függvény. A függvény paramétereként megadott *list* érték a későbbiekben elkészített *list* nevű listára utal. Ennek segítségével van összehangolva a szövegdozst és a lista, tehát ha gépelünk, akkor az optimálisan keresi a listában a megfelelő értéket.

A keresést feldolgozó gomb megalkotásához egy gomb típusú *input* elemet kellett használnom:

```
<input type="button" value="Keress!"  
onclick="window.parent.setVP(this.form.list.value,1000000);"/>
```

Az *onclick* paraméter értékeként a *setVP* függvény nevét adtam meg. Ennek köszönhetően, ha valaki rákattint a gombra, akkor működésbe lép a függvény, és a listában kiválasztott településhez rendelt koordináták alapján a gömböt a megfelelő helyre forgatja. A *setVP* függvénybe írt 1 000 000 érték méterben értendő, és azt jelenti, hogy a forgás végére a függvény ilyen távolságra fogja beállítani a szemmagasságot.

Legvégül a településneveket tartalmazó listát készítettem el. Ezt egy `<SELECT>` elnevezésű HTML kód segítségével hoztam létre. Az utasítás alapvetően egy üres listát készít, amit valamilyen módon fel kell tölteni elemekkel. A feltöltéséhez az `<OPTION>` elemet használtam:

```
<select>  
  <option value="47.542221,16.655038">Sopron</option>  
  <option value="48.050572,16.313855">Bécs</option>  
  <option value="47.934364,20.311349">Eger</option>  
  <option value="47.551166,21.540019">Debrecen</option>  
</select>
```

A parancsnak kétféle értéket adtam meg. Az egyik értéket az elem *value* attribútuma után írtam, ami a földrajzi koordinátákat foglalja magába. Ez kerül elküldésre a szervernek. A másik értéket az `<OPTION>` utasítás nyitó és záró eleme közé írtam. Az itt feltüntetett szöveg kerül megjelenítésre a böngésző által a listában.

Mint látható, ennek az eljárásnak az a hátránya, hogy minden egyes elemet kézzel kell beírni, és ezres nagyságrendeknél ez már szinte megvalósíthatatlan. Ráadásul az én esetemben nem csak a települések nevét, hanem azoknak a földrajzi koordinátáit is be kell gépelni. Emiatt találtam ki azt a módszert, hogy a meglévő adatokat egy MySQL adatbázisban összegzem, majd egy ciklus használatával feltöltöm velük a listát. Véleményem szerint ez a lépés rendkívül megkönnyítette a munkámat, ugyanis enélkül, ennyi adattal nehéz lett volna dolgozni. A következő részben részletesebben ismertetem a MySQL adatbázis létrehozását, és az adatok meghívásának a módszerét a *kereso.php* állományomból.

4.3.2.2. MySQL adatbázis

Ahhoz, hogy az összegyűjtött adatokat valamilyen módon összekapcsoljam a Google Earth beépülő modullal, létrehoztam egy MySQL adatbázist. A MySQL (szabványos lekérdezőnyelv) egy nyílt forráskódú relációs adatbáziskezelő-rendszer, amelynek több millió felhasználója van a világon. Népszerűségét igazolja, hogy rengeteg nagyvállalat használja adatbázismotorként nagy forgalmú honlapjához. Segítségével dinamikus oldalakat hozhatunk létre és különböző lekérdezéseket hajthatunk végre adatbázisunkban. (MELONI 2002)

A MySQL adatbázis létrehozásához a phpMyAdmin szoftvert használtam. A phpMyAdmin egy PHP kódban írt nyílt forráskódú eszköz, amelynek segítségével a MySQL-t interneten keresztül tudjuk kezelni. A program képes létrehozni új adatbázisokat, táblákat, amelyeket később módosíthatunk vagy törölhetünk. Módosítás alatt újabb mezők hozzáadását, kezelését értem.

A programmal létrehoztam egy adatbázist *gombindex* néven, azon belül pedig egy táblát, amit *perczelnek* neveztem el. A *perczel* táblában öt mezőt készítettem. A *lon* mező a hosszúsági értékek, a *lat* mező a szélességi értékek, a *leolv_nev* a gömbről leolvasott nevek, a *valsz_nev* pedig a feltételezett nevek tárolására szolgál. Ezek mellett létrehoztam még egy *id*, azonosítókat tároló mezőt is, amit a program automatikusan készít el oly módon, hogy minden egyes sorhoz egy-egy sorszámot rendel, és ezáltal megkülönbözteti őket egymástól.

A mezők feltöltéséhez a korábban elkészített Excel tábla adatait használtam fel. Első lépésként elmentettem a táblát .csv kiterjesztéssel, mert az egyes adatbáziskezelő-rendszerek, mint például a MySQL is, nem tudják olvasni az Excel alapformátumát, az .xls-t, csak ezt az általa egyébként egyszerűen létrehozható formátumot. A .csv (Comma Separated Values) kiterjesztésű állományok lényegében ugyanazokat az adatokat tartalmazzák, mint egy egyszerű .xls fájl, azzal a különbséggel, hogy az egyes oszlopokat vesszővel választják el, és minden új adatsoruk egy új sort reprezentál a táblázatban.

Az Excel táblában levő oszlopokat és a MySQL adatbázisban elkészített *perczel* tábla mezőit azonos módon neveztem el. A .csv fájl *gombindex* adatbázisba való importálásához ez szükséges lépésnek bizonyult, mivel a phpMyAdmin lehetőséget ad arra, hogy így relációt hozzak létre a két tábla között. Az importáláskor megadtam, hogy milyen karakter jelölje az egyes mezők végét (pontosvessző), valamint beállítottam,

hogy a forrás táblából az első sor elemeit értelmezze az oszlopok neveinek. Amikor végrehajtottam az importálást, a program relációt állított fel a két tábla között, amelynek következtében a *perczel* tábla mezőit feltöltötte a .csv tábla oszlopainak az elemeivel. Így lényegében elkészült az adatbázisom, amellyel a későbbiekben dolgozhattam.

	lon	lat	leolv_nev	valsz_nev	id
<input type="checkbox"/>	16.3139	48.0506	Bécs	Bécs	1
<input type="checkbox"/>	16.4404	47.7926	U????	?	2
<input type="checkbox"/>	16.655	47.5422	Sopron	Sopron	3
<input type="checkbox"/>	14.6242	46.0506	Laibach	Laibach	4
<input type="checkbox"/>	14.5061	47.9555	Steyr?	Steyr	5
<input type="checkbox"/>	15.4113	47.1683	Gratz	Gratz	6
<input type="checkbox"/>	14.1283	48.1759	Linz	Linz	7
<input type="checkbox"/>	14.5219	48.7033	Budweis	Budweis	8
<input type="checkbox"/>	15.5215	49.4432	Jglau	Iglau	9

4. ábra - Részlet a MySQL adatbázisból

4.3.2.3. Csatlakozás az adatbázishoz

A MySQL adatbázis webes felületének megvalósításához a PHP nyelvet használtam. A PHP (Hypertext Preprocessor) egy nyílt forráskódú kiszolgálóoldali parancsnyelv, amelyet általában HTML oldalakon alkalmaznak dinamikus weblapok készítésére. Számos lehetősége mellett végrehajthatunk vele adatbázis-lekérdezéseket, valamint kapcsolatba léphetünk távoli kiszolgálókkal. A PHP kódoknak `<?php` és `?>` elemek között kell elhelyezkedniük és a kódot tartalmazó fájl kiterjesztésének .php-nak kell lennie, ugyanis a kiszolgáló PHP-értelmező motorja ennek segítségével tudja felismerni. A fájl a PHP kódok mellett tárolhat HTML kódokat is. Ilyenkor a PHP parancsokat a kiszolgáló nem küldi el, viszont a programjainkban lévő HTML elemek érintetlenül maradnak, tehát a felhasználó ugyanúgy elérheti őket. (ZANDSTRA 2004)

A MySQL-hez való kapcsolódáshoz, a lekérdezés végrehajtásához és a neveket tartalmazó lista feltöltéséhez a következő kódsorozatot alkalmaztam a PHP nyitó és záró elemei között:

```
$db=mysql_connect('localhost','felhasználói név','jelszó');
mysql_select_db('gombindex',$db);
$eredmeny=mysql_query("select * from perczel where valsz_nev like
```



```
'%{$_POST['minta']}%' and valsz_nev not like '%%?%%' and valsz_nev<>' order
by valsz_nev", $db);

while ($val=mysql_fetch_assoc($eredmeny))
print "<option value=".$val['lat'].",".$val['lon'].>".$val['valsz_nev'];
print "</option>";

mysql_close($db);
```

Ezt a PHP kódsorozatot a korábbiakban említett <SELECT> utasítás nyitó és záró elemei közé helyeztem el, hogy feltöltsen településnevekkel az általa létrejövő listát. A parancs nyitó eleme a következőképpen néz ki:

```
<select name="list" size="30"
ondblclick="window.parent.setVP(this.value,1000000)">
```

Az *ondblclick* paraméter arra szolgál, hogy a lista egyik elemére való dupla kattintás után, a településhez rendelt koordinátapár és a *setVP* függvény segítségével a gomb odaforogjon a megfelelő helyre.

Az adatbázis-szerverhez való kapcsolódást a *mysql_connect* függvény segítségével tudtam elérni, melynek három paramétereként megadtam a gépnevet, a felhasználónevet és a jelszót. A függvény ellenőrzi az általunk megadott adatok hitelességét, majd egy kapcsolat indexel tér vissza, amit egy változóban tárol (*\$db*).

Második lépésként kiválasztottam azt az adatbázist, amelyből ki szerettem volna nyerni az adatokat. Ezt a *mysql_select_db* függvény használatával oldottam meg, amelyben megadtam az adatbázis nevét (*gombindex*), illetve a *\$db* változót, ami az előző lépésben tárolta a kapcsolat indexet. (MELONI 2002)

A *mysql_query* függvénnyel végrehajtottam egy lekérdezést, amelynek segítségével kiválasztottam az összes olyan elemét a *perczel* tábla *valsz_nev* oszlopának, ami tartalmaz valamilyen adatot, de nincsen benne háromnál több kérdőjel egymás mellett. A lekérdezés eredményét az *\$eredmeny* változóban tároltam.

A következő lépésben egy WHILE ciklus és a *mysql_fetch_assoc* függvény segítségével végighaladtam az *\$eredmeny* változó elmein, és az egyes mezők értékét egy újabb változóban rögzítettem. Ezt a változót *\$val*-nak neveztem el. Ezután a cikluson belül elhelyeztem a már ismertetett <OPTION> parancsot. A parancs *value* attribútumaként a *\$val* változó *lat* és *lon* értékeit írtam be, valamint az <OPTION> nyitó és záró elemei közé ugyanennek a változónak a *valsz_nev* értékét használtam. Ezek az értékek megegyeznek a MySQL adatbázis mezőivel, ennek köszönhetően mindig az adott mező elemeiből használ fel adatokat. A ciklus lefutásakor egymás után elkészíti az

<OPTION> parancsokat, feltöltve azokat a településnevekkel, és a hozzájuk tartozó koordinátapárokkal. Mindeközben a listát is folyamatosan feltölti, amíg el nem fogynak az adatbázisban a településnevek. Végül a kapcsolat lezárásához a *mysql_close* függvényt alkalmaztam.

4.3.3. Navigációs és kisegítő eszközök

Az utolsó <DIV> elembe, egy táblázatba foglalva olyan eszközöket készítettem el, amelyek általában már ismertek a Google Earth alapszoftverből. Úgy hoztam létre őket, hogy a honlap betöltésekor kikapcsolt állapotban legyenek, viszont igény szerint bármikor be lehet őket kapcsolni. Ezekhez az eszközökhöz tartozó funkciók:

- Állapotsor
- Navigációs eszközök
- Fokhálózat
- Aránymérték

Mellettük még két másik gombot is elkészítettem. Az egyik a településekhez tartozó helyjelzők ki- és bekapcsolására szolgál, tehát megszüntethetjük vele a helyjelzők megjelenítését a gömbön, illetve fordított esetben meg is jeleníthetjük őket. A másik gombot a helyjelzőket tartalmazó KML fájl kódjának a generálásához használtam. Ezt tulajdonképpen a munkám megkönnyítése érdekében hoztam létre, az elkészült honlapban nem szerepel.

4.3.3.1. A navigációs eszközök elkészítése

A navigációs eszközök megvalósításához szükségem volt egy JavaScript függvény használatára:

```
function updateOptions() {
    var options = ge.getOptions();
    var form = document.options;

    options.setStatusBarVisibility(form.statusbar.checked);
    options.setGridVisibility(form.grid.checked);
    options.setScaleLegendVisibility(form.scaleLegend.checked);

    if (form.nav.checked) {
ge.getNavigationControl().setVisibility(ge.VISIBILITY_SHOW);
    } else {
ge.getNavigationControl().setVisibility(ge.VISIBILITY_HIDE);
    }
}
```

Ez a függvény arra ad utasítást, hogy ha az egyes eszközök ki- és bekapcsoló gombja, azaz checkboxa, be van kapcsolva, akkor az eszköz funkciói megjelenítésre kerüljenek a modulban. Ha a checkbox nincs bekapcsolva, akkor értelemszerűen nem jeleníti meg őket.

A ki- és bekapcsoló gombokat a <DIV> tároló elemen belül egy <FORM> tagben készítettem el, ami arra szolgál, hogy információt küldjön a szervernek. Létrehozásukhoz *checkbox* típusú <INPUT> elemeket használtam, amelyeknek az *onclick* attribútumainál az *updateOptions* elnevezésű függvényt hívtam meg. Ennek köszönhetően, ha rákattintunk a gombokra, azaz ha be- vagy kikapcsoljuk őket, akkor működésbe lép a függvény. Az eszközök funkcióit és tulajdonságait majd az elkészült honlap bemutatásánál tárgyalom részletesebben.

4.3.3.2. A helyjelzők elkészítése

A helyjelzők, vagy más néven placemarkok, létrehozásához hasonló eljárást alkalmaztam, mint a keresési lista feltöltéséhez. Elkészítettem egy PHP fájlt, amelyben a már meglévő MySQL adatbázisból egy ciklus segítségével kinyertem az adatokat. Erre a lépésre azért volt szükségem, mert minden egyes településhez el kellett készítenem egy-egy helyjelzőt, amit rendkívül hosszadalmas munka lett volna a ciklus véghezvinni.

Első lépésként létrehoztam egy *placemark.php* fájlt, amit feltöltöttem a meghíváshoz használt PHP kódokkal, illetve a később elkészített KML fájlhoz szükséges egyéb utasításokkal. Az adatok meghívását az adatbázisból majdnem ugyanúgy végeztem el, mint a lista esetében, azzal a különbséggel, hogy a WHILE cikluson belül más utasításokat adtam meg:

```
while ($val=mysql_fetch_assoc($eredmeny))
print
"<Placemark><name>".$val['valsz_nev']."</name><description><![CDATA[A település leolvasott neve:<font color="red"> ".$val['leolv_nev']."</font><br />A település valószínű neve:<font color="red"> ".$val['valsz_nev']."</font><br />A település koordinátái (fi,lambda):<font color="red"> ".$val['lat']." , ".$val['lon']."</font>]]></description><styleUrl>#m_ylw-pushpin</styleUrl><Point><coordinates>".$val['lon']." , ".$val['lat']." , 0</coordinates></Point></Placemark><br />";
```

A kódsorozat azért lett ilyen rendkívül hosszú és bonyolult, mert sok, a HTML nyelvben speciálisnak számító karakterek tartalmaz. Ezeket a speciális karaktereket nem írhattam

be a parancsba az eredeti alakjukban, mert akkor a böngésző egy utasítás részeként értelmezte volna őket. Helyettük több karakterből álló kódokat kellett alkalmaznom, amelyeket a következő táblázatban tüntetek fel:

Speciális karakterek	
<	<
>	>
"	"

A relációs jelek szinte minden esetben az utasítások nyitó és záró címkéjéül, a gondolatjelek pedig az értékek megadásánál használatosak. A hosszú kódsorozatnak a PHP fájl lefutása után, egy példán keresztül bemutatva, a következő lesz az eredménye:

```
<Placemark>
  <name>Bécs</name>
  <description><![CDATA[A település leolvasott neve:<font color="red">
Bécs</font><br />A település valószínű neve:<font color="red"> Bécs</font><br
/>A település koordinátái (fi,lambda):<font color="red"> 48.0506 ,
16.3139</font>]]>
  </description>
  <styleUrl>#m_ylw-pushpin</styleUrl>
  <Point>
    <coordinates>16.3139,48.0506,0</coordinates>
  </Point>
</Placemark>
```

Mivel a bonyolultabb kódsorozat egy cikluson belül lett elhelyezve, ezért annyiszor fut le, ahány településnév ment át az előre meghatározott feltételeken. Ezeket a feltételeket a MySQL lekérdezéskor szabtam meg, ugyanúgy, mint a listánál. A lefutás befejezésekor mindegyik településhez generál egy kódsorozatot, ami a hozzá tartozó helyjelző létrejöttéért felelős. Amellett, hogy létrehozza a helyjelzőket, fel is tölti őket a településekre vonatkozó adatokkal.

Az eredményben látható <PLACEMARK> utasítás hatására készül el a helyjelző, s ezen belül további fontos utasítások is találhatóak. A <NAME> utasításban levő rész adja meg a helyjelző nevét, ami a gömbön is fel van tüntetve a helyjelző mellett. Ez esetben megegyezik a településnévvel. A <DESCRIPTION> utasítás segítségével lehet feltölteni a helyjelzőt információkkal. Itt megadtam a település leolvasott és valószínűsített nevét, illetve a koordinátáit. Az értékeket vörös színnel írtam ki. A helyjelzőre vonatkozó információk megtekintéséhez az ikonjára való kattintás után van lehetőség. Ilyenkor egy buborék formájában jeleníti meg a rá vonatkozó adatokat. Végül ahhoz, hogy a helyjelző a településnek a gömbön feltüntetett

helyén helyezkedjen el, a <COORDINATES> utasítást kellett használnom. Ehhez azt kellett tennem, hogy az utasítás belsejében megadom a település földrajzi koordinátáit vesszővel elválasztva. Lehetőségem lett volna arra is, hogy magassági adatot is megadjak, de erre nem volt jelen esetben szükségem, ezért ezt az értéket nullára állítottam.

A *placemark.php* fájlom ezek mellett tartalmaz még a KML fájl létrejöttéhez szükséges egyéb utasításokat is. Ezeket a Google Earth szoftver segítségével készítettem el. Itt találhatóak többek között a helyjelzőkhöz tartozó ikonok stílusára, méretére, illetve színére vonatkozó utasítások is. Ennél a résznél szintén ügyelnem kellett az utasításokban található speciális karakterekre.

4.3.3.3. KML kódgeneráló gomb

Miután elkészültem a PHP fájlal, a honlap utolsó <DIV> elemében készítettem egy gombot, aminek az lett a funkciója, hogy megnyomása után lefut a PHP fájl. Ezt egy <FORM> utasításon belül valósítottam meg egy egyszerű *input* paranccsal. A gomb megnyomásakor a böngésző egy új lapon jeleníti meg a fájl hatására generált adatokat. Ez már tartalmazza az összes településhez tartozó helyjelzőhöz szükséges kódot. Innentől kezdve a helyjelzőkért felelős KML fájl létrehozásához még annyi volt a teendőm, hogy a generált kódsorozatot átmásoltam egy olyan programba, ami képes KML fájlokat elkészíteni. Erre a feladatra kiválóan alkalmas a Notepad++ szoftver, amelyben a bemásolt kódokat tartalmazó dokumentumot el kellett mentenem a megfelelő kiterjesztéssel.

4.3.3.4. Helyjelzők ki- és bekapcsolására szolgáló gomb

Az előbb említett <DIV> elem utolsó részeként egy olyan gombot készítettem el, ami a KML fájlnak, azaz a helyjelzők megjelenítésének a ki- és bekapcsolására szolgál. Ennek a fájlnak a betöltését a modulba az *initCB* alapvető függvényen belül végeztem el, hasonló módon, mint a Perczel-gömb kinézetéért felelős KML fájl betöltését:

```
var url = 'http://mercator.elte.hu/~totqaat/placemarks.kml';  
google.earth.fetchKml(ge, url, kmlFinishedLoading);
```

A ki- és bekapcsolás működéséhez az alábbi függvények forráskódba való beírására volt szükségem:

```
function kmlFinishedLoading(obj) {
    kmlObject = obj;
    if (kmlObject) {
        if ('getFeatures' in kmlObject) {
            ge.getFeatures().appendChild(kmlObject);
        }
        if (kmlObject.getAbstractView())
            ge.getView().setAbstractView
            (kmlObject.getAbstractView());
    }
}

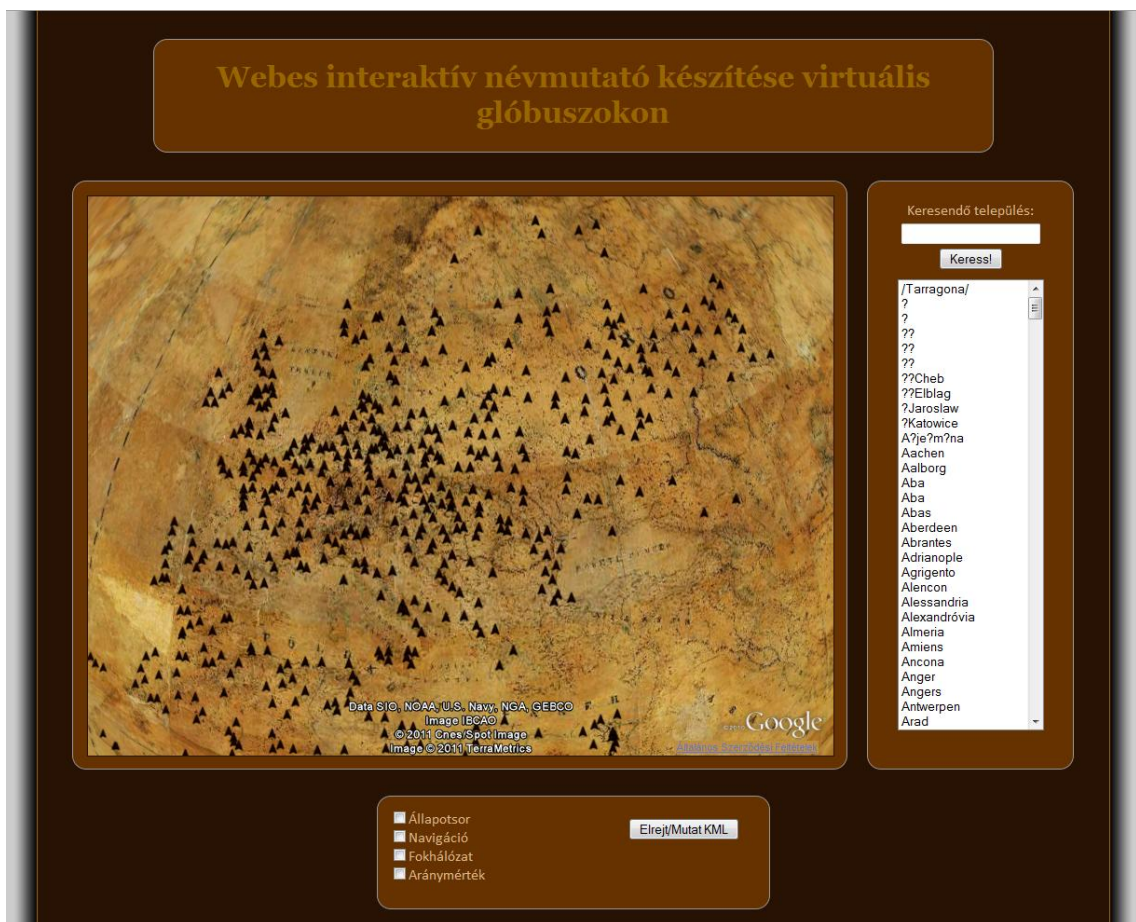
function showHideKml() {
    kmlObject.setVisibility(!kmlObject.getVisibility());
}
```

A két függvény közül a *showHideKml* elnevezésűt kellett meghívnom a honlap törzsében a funkció életre keltéséhez. Ezzel a lépéssel az volt a célom, hogy a gomb megnyomásakor ennek a függvénynek a segítségével eltűnjenek a helyjelzők a gömbről, majd újbóli megnyomásra megjelenjenek. Ezt a gyakorlatban úgy vittem véghez, hogy a gombhoz használt <INPUT> elem attribútumaként megadtam egy *onclick* utasítást, aminek értékeként a függvény nevét írtam be. Ennek hatására már megfelelően működött a gombom.

5. A honlap bemutatása

Miután az összes szükséges fájl elkészült, felmásoltam őket a tanszéki tárhelyemre. Így a weblapom már bárholnan elérhető az interneten keresztül.

A honlap megtekintéséhez a következő webcím begépelése szükséges a böngészőbe: <http://mercator.elte.hu/~totqaat>. Ha ez megtörtént, akkor az alábbi kép fogadja a látogatót:



5. ábra - A webes interaktív névmutató

A fényképekből összeállított KMZ állományt egyes esetekben lassabban tölti be Google Earth beépülő modul. A betöltés sebességének csökkenése a fájl viszonylagos nagy mérete miatt lehetséges (a mérete kb. 130 MB). Miután a betöltés megtörtént, elkezdhetjük a honlap böngészését.

5.1. Nézőpont változtatás

Lehetőség van a glóbusz vizsgálatára közelebbi nézőpontból. Az egér görgőjének segítségével lehet ráközelíteni a gömbre, illetve el is lehet azt forgatni bármilyen irányba. Ezek a működési sajátosságok sokak számára ismertek lehetnek a Google Earth szoftverből. A közelítésnek két előnye van. Az egyik az, hogy bizonyos távolságból megjelennek a helyjelzők mellett a hozzájuk rendelt településnevek. Az alap nézőpontból, amit automatikusan betöltött a honlap, azaz 4 500 000 méter szemmagasságból azonban még nem láthatóak. A másik előnye, hogy lehetőség van a KMZ fájlban levő nagyobb felbontású képek segítségével részletesebben vizsgálni a gömböt, mint ahogyan alapállapotban tudnánk. Ilyenkor ugyanis még csak egy kevésbé részletes, tehát kisebb felbontású képet tölt be a honlap és csak egy bizonyos mértékű közelítés után tölti be a nagyobb felbontású képeket. Közelítés után a következő képet láthatjuk:



6. ábra - A közelítés utáni nagyobb felbontású kép

Mint látható, ekkora távolságból már sokkal jobban lehet szemlélni a gömb sajátosságait. Szépen kitűnik a vízrajz, a partvonal, és az országhatár is, ellenben a névanyag még kevésbé látható.

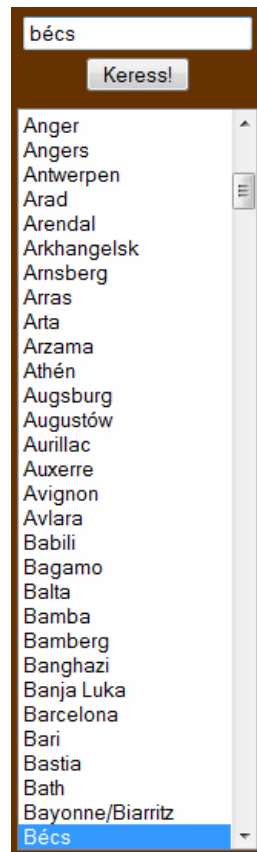
Ha ennél is részletesebben szeretnénk vizsgálni a glóbuszt, akkor lehetőség van még nagyobb mértékben való ráközelítésre. Ilyenkor már a településnevek és a vízrajzi nevek is tökéletesen látszanak. Ez természetesen csak azokra a nevekre igaz, amik nem koptak le annyira az idők folyamán, hogy ne lehetne olvasni őket. Az ilyen mértékű közelítés után az alábbi képen látható egy példa a nevek olvashatóságára:



7. ábra - Példa a nevek olvashatóságára

5.2. A névmutató használata

A névmutató használata is rendkívül egyszerű. Egy településnév megkereséséhez a korábban említett keresési elemek használata szükséges. Kétféle módon kereshetünk. Az egyik, hogy a szövegdobozba beírjuk a keresendő település nevét, a másik pedig az, hogy a listában böngészünk a településnevek között. Ha a szövegdobozba írtuk be a nevet, akkor a lista is odaugrik a megfelelő településhez a következőképpen:

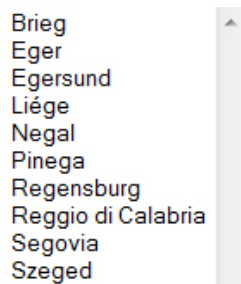


8. ábra - A névmutató

Mindezek után, ha azt szeretnénk, hogy a honlap a gömbön is megmutassa a keresett települést, akkor újfent két dolgot tehetünk. Szövegdobozba beírás esetén lehetőség van arra, hogy a „Keress!” feliratú gomb megnyomásával működésbe hozzuk a kívánt funkciót. Ellenben, ha a listát használtuk a település megkereséséhez, akkor dupla kattintás szükséges a keresett település nevére.

Ha a település nevének csak egy részét ismerjük, aminek nem feltétlenül kell a szó elejének lennie, akkor lehetőség van arra is, hogy a meglévő szótöredék alapján keressünk az adatbázis nevei között. Ilyenkor a szövegdobozba be kell írni ezt a szótöredéket, majd az Enter gomb megnyomása után a honlap kiírja a listában az összes

olyan településnevet, amiben szerepel a keresett rész. Például az „eg” kifejezés beírása után az alábbi eredményt kaptam a listában:



Brieg
Eger
Egersund
Liége
Negal
Pinega
Regensburg
Reggio di Calabria
Segovia
Szeged

9. ábra - A leszűkített lista

Mint látható, mindegyik településnévben szerepel az előbb említett kifejezés, tehát a keresés sikeres volt. Ezek után már könnyebben kiválaszthatjuk a leválogatott települések között a számunkra megfelelőt.

Ha a felhasználó az előbb felajánlott nevek közül például Egert szeretné megtekinteni a glóbuszon, akkor egy dupla kattintás után a gömb odaforog a megfelelő helyre. Ezután tetszőlegesen rá lehet közelíteni a településre, egészen addig, hogy a gömbi megírása is olvasható legyen. Szükség esetén a település helyjelzőjére kattintva a honlap információkat ír ki az adott település földrajzi koordinátáiról, valamint a kigyűjtéskor leolvasott névről, illetve a feltételezett településnévről egy buborék formájában:



10. ábra - Információk a településről

Eger esetében a két név pont megegyezik, de számos esetben a leolvasott névben több kérdőjel is szerepelhet a nem megfelelő olvasási feltételek miatt. Előfordulhat olyan is, hogy a valószínűsített névnél levő rész üres, mert az adott településhez nem találtak a kigyűjtésnél a forrásokból megfelelő adatot.

5.3. A navigációs és kisegítő eszközök használata

Ezek a korábbiakban már említett eszközök sok esetben hasznosak lehetnek a gömb vizsgálata során. Nagy részük az alapszoftverben is széles körben alkalmazott. Összességében négy ilyen eszközt készítettem el, illetve közéjük helyeztem a helyjelzők kapcsoló gombját is.

Az „Állapotsor” elnevezésű eszköz bekapcsolásakor a Google Earth modul alsó részében megjelenik egy szürke színű, enyhén átlátszó sáv, ami különböző értékeket tartalmaz:



11. ábra - Az állapotsor

Mint látható, megjelennek földrajzi koordináta értékek, egy magassági érték, ami mindig nulla, ugyanis korábban ezt így állítottam be, valamint mutatja a szemmagasságot is kilométerben kifejezve. A koordináták mindig a kurzor helyzetét jelentik a gömbön.

A navigációs gomb bekapcsolásával három különböző eszköz jelenik meg a modul jobb felső sarkában:



12. ábra - Navigációs eszközök

A két kör alakú eszköz közül a felsővel változtathatjuk a gömbre való rálátás szögét, valamint az északtól való eltérést. Az alsó kör alakú eszközzel végigpásztázhatunk a gömbön. Ezt a funkciót az egér bal gombjának a gömbön való nyomva tartásával, illetve a kurzor mozgatásával is elérhetjük. A legelső eszköz segítségével változtathatjuk a szemmagasságot. Szükség szerint közelebbi, illetve távolabbi nézőpontból szemlélhetjük a gömböt. A ráközelítés akkor lehet különösen hasznos, ha a glóbuszt odaforgattuk a keresési funkciók valamelyikével a megfelelő településhez, de a szemmagasságot még mindig túl távolinak találjuk. Ilyenkor a csúszka segítségével oly módon közelíthetünk rá jobban a településre, hogy nem mozdul el a gömb az általunk vizsgált területről, míg az egér görgőjének a használatánál ez nem valósítható meg ennyire pontosan.

A harmadik gomb bekapcsolása után egy fokhálózat jelenik meg a gömbön:

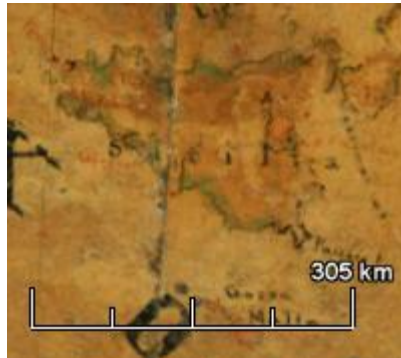


13. ábra - A fokhálózat különböző nagyítási szinteken

Ahogy az ábrán is látható, a fokhálózat sűrűsége nem állandó. Ha távolabbi nézőpontból vizsgáljuk a gömböt, akkor ritkább a beosztása, ha viszont ráközelítünk, akkor sűrűbb. Ez folyamatosan változik a közelítés, és a távolodás mértékétől függően.

Az utolsó gomb bekapcsolásakor egy aránymérték jelenik meg a modul bal alsó sarkában. A szemmagasság változtatásától függetlenül az aránymérték beosztása

állandó, csupán az általa reprezentált távolság nagysága változik. Ennek a távolságnak a nagysága folyamatosan látható az aránymérték jobb felső sarkában. Ez az érték kilométerben, illetve nagyobb mértékű ráközelítés esetén, méterben van kifejezve az alábbi módon:



14. ábra - Az aránymérték

Ezen eszközök használatának a vegyítésével a felhasználó olyan szemszögből szemlélheti a gömböt, ahogyan az számára a legkedvezőbb. Így nem kizárható, hogy sok esetben, kihasználva a háromdimenziós számítógépes ábrázolás nyújtotta előnyöket, részletesebb képet kaphat, mint az eredeti gömb vizsgálatakor.

6. Összefoglalás

Munkám során véleményem szerint sikerült megvalósítani mindazt, amit eredetileg célként kitűztem. Elkészítettem egy olyan webes névmutatót Perczel László glóbuszához, ami része lehet a kulturálisérték-mentés folyamatának. Érdemesnek tartottam többet foglalkozni ezzel a nagyszerű műalkotással, ily módon megőrizni annak értékeit, s bebizonyítani, hogy fontos szerepet játszik életünkben az értékmentésnek ez a formája.

Dolgozatom első felében bemutattam a tanszék keretein belül elkészült Virtuális Glóbuszok Múzeumát és a múzeum részét képező Perczel-glóbuszt, illetve annak alkotóját. Azért tartottam fontosnak a glóbuszmúzeum keretein belül folyó gömbök digitalizálásának az ismertetését, mert végeredményben a névmutató elkészítéséhez is ez a munka szolgáltatta az alapötletet.

Munkám másik, tartalmasabb részében a webes névmutató megvalósításának a folyamatait, majd az elkészült honlap használatához szükséges ismereteket mutattam be. Részletesen leírtam a honlapkészítéshez használt informatikai nyelvek alapismereteit, valamint azok használatát egy ilyen munka elkészítése során. Kitértem a Google Earth beépülő modul honlapokon való alkalmazásának a technikai megoldására, valamint mindezek mellett ismertettem az egyik legnépszerűbb adatbáziskezelő-rendszert, a MySQL-t, illetve annak használatát az általam létrehozott adatbázison.

Annak ellenére, hogy a honlap nem lett bonyolult, bemutattam funkcióinak a használatát és a működését példákon keresztül, hogy annak számára is egyértelmű legyen, aki értékeli az értékmentésnek ezt a formáját, tehát műalkotásnak számító gömbök virtuális megőrzését, de nem feltétlenül van otthon az informatika világában.

7. Köszönetnyilvánítás

Ezúton szeretném megköszönni témavezetőmnek, Gede Mátyásnak, hogy mindig készségesen segített munkám során. Valamint szeretnék köszönetet mondani még Márton Mátyásnak, amiért hasznos információkkal látott el Perczel Lászlóról és földgömbjéről, illetve Nemes Zoltánnak a glóbuszról készült fényképért (1. ábra).

8. Irodalomjegyzék

Nyomtatott irodalom

- **Ambrus-Fallenbüchl Zoltán, 1963:** Magyarország legnagyobb földgömbje száz éves, In: Geodézia és Kartográfia, XV. évf., 1963/1, pp. 61-62
- **Dr. Tímár Gábor, 2007:** A ferrói kezdőmeridián, In: Geodézia és Kartográfia, LIX. évf., 2007/12, pp. 3–7.
- **Márton Mátyás, 2008:** Egy elfelejtett magyar csoda: Perczel László földgömbje – az első „világtérképű”? In: Geodézia és Kartográfia, LX. évf., 2008/3, pp. 9–16.
- **Márton Mátyás, 2009:** A Virtuális Glóbuszok Múzeuma – Interneten elérhető szemléltetőeszköz. In: Pajtókné Tari I., Tóth A. (szerk.), Változó Föld, változó társadalom, változó ismeretszerzés: tudományos konferencia az EKF Földrajz Tanszék 60 éves jubileumához kapcsolódva, pp. 396-404.
- **Meloni, Julie C., 2002:** Tanuljuk meg a MySQL használatát 24 óra alatt, Kiskapu kiadó, Budapest.
- **Moncur, Michael, 2002:** Tanuljuk meg a JavaScript használatát 24 óra alatt, Kiskapu kiadó, Budapest.
- **Nyuli Éva, 2009:** Európa településeinek térinformatikai adatbázisa a Perczel gömb alapján. Szakdolgozat, ELTE IK Térképtudományi és Geoinformatikai Tanszék, Budapest.
- **Tóth Bettina, 2009:** Dél-Amerika településneveinek geoadatbázisa a Perczel-gömb alapján. Szakdolgozat, ELTE IK Térképtudományi és Geoinformatikai Tanszék, Budapest.

- **Való Adrienn, 2009:** Afrika településneveinek geodatbázisa a Perczel-gömb alapján. Szakdolgozat, ELTE IK Térképtudományi és Geoinformatikai Tanszék, Budapest.
- **Zandstra, Matt, 2004:** Tanuljuk meg a PHP 5 használatát 24 óra alatt, Kiskapu kiadó, Budapest.

Internetes hivatkozások

- **Abonyi-Tóth Andor, 2000:** HTML tananyag – Speciális karakterek, ékezetes betűk.
http://www.aut.bmf.hu/konya/okt/netint_www/htmloffline/htmltananyag/gyak1/apok.htm
- **Bártházi András, 2004:** CSS alapjai I.
<http://weblabor.hu/cikkek/cssalapjai/>
- **Google, 2011:** Google Earth API Developer's Guide.
<http://code.google.com/intl/hu-HU/apis/earth/documentation/>
- **Google, 2011:** Google Föld fejlesztők számára.
<http://www.google.com/intl/hu/earth/developers/>
- **Google, 2011:** KML Reference.
<http://code.google.com/intl/hu-HU/apis/kml/documentation/kmlreference.html>
- **W3C, 1999:** HTML 4.01 Specification
<http://www.w3.org/TR/html401/>
- **HTML Info:** Blokk szintű elemek.
<http://htmlinfo.hu/html-4/blokk szintu-elemek/>
- **Keyhole Markup Language, 2010**
http://hu.wikipedia.org/wiki/Keyhole_Markup_Language

- **phpMyAdmin devel team, 2011:** About phpMyAdmin.
http://www.phpmyadmin.net/home_page/index.php
- **Vámosi Gergő, 2008:** Böngészőből is használható a Google Earth.
<http://www.origo.hu/techbazis/internet/20080529-bongeszobol-is-hasznalható-a-google-earth.html>

9. Nyilatkozat

Alulírott **Tokai Tibor** (ETR azonosító: **TOTQAAT**) a **Webes interaktív névmutató készítése virtuális glóbuszokon** című szakdolgozat 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 témavezető által benyújtásra elfogadott szakdolgozat elektronikus publikálásához (PDF formátumban a tanszéki honlapon)

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest,.....

.....
a hallgató aláírása