EÖTVÖS LORÁND TUDOMÁNYEGYETEM TERMÉSZETTUDOMÁNYI KAR

Webes összehasonlító névrajzi kereső kialakítása glóbuszokhoz

SZAKDOLGOZAT FÖLDTUDOMÁNYI ALAPSZAK TÉRKÉPÉSZ ÉS GEOINFORMATIKUS SZAKIRÁNY

Készítette: Beszkid Arthur

Témavezető: Ungvári Zsuzsanna tanársegéd ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2017

Tartalomjegyzék

1. Bevezetés	
2. Glóbuszok Magyarországon	4
2.1. A magyarországi glóbuszkészítés történelmi áttekintése	4
2.2. Virtuális földgömbök	5
2.3. Virtuális Glóbuszok Múzeuma	5
3. A felhasznált adatbázis felépítése	6
3.1. A dolgozatom előzménye: Szabó Virág diplomamunkája	6
3.2. Shapefile	7
3.3. A tengerneveket tartalmazó tábla előállítása	9
3.4. A glóbuszokon előforduló neveket tartalmazó tábla létrehozása	
3.5. A gömbök leíró adatait tartalmazó táblázat	
4. A weboldal elkészítése	
4.1. USBWebserver	13
4.2. phpMyAdmin	14
4.3. Az adatbázis létrehozása és feltöltése	14
4.4. A weboldal "külleme"	16
4.5. A CSS formázás	16
4.6. A megjelenítő	17
4.6.1. Glóbusz textúrák	
4.6.2. A tengerhatárokat tartalmazó fájl átalakítása a webes keresőhöz	
4.6.3. A KML fájl készítésénél felmerült nehézségek	
4.7. A HTML-kód	24
4.8. A PHP-kód	
4.9. Weboldal elérhetősége	
5. Összegzés	
Köszönetnyilvánítás	
Irodalomjegyzék	

1. Bevezetés

Földünk bemutatására az egyik legszemléletesebb eszköz a földgömb: amely kicsinyítve, ugyan, de torzítás mentesen, háromdimenzióban ábrázolja a Földfelszínt (vagy bármely más égitestet) (Klinghammer, 1998). Korábban a földrajz oktatásában is jelentősebb szerepet játszott, mint manapság: a régi iskolai eszközök elkoptak, és többnyire nem pótolták őket.

Az internetes térképi alkalmazások széles körű elterjedésével a papír alapú térképi tájékozódás egyre kisebb szerepet tölt be életünkben: a mindennapokban, vagy akár az oktatásban és a kutatásban egyaránt. Ezért fontosnak tartom, hogy ezzel lépést tartva, a számunkra fontos értékeket digitalizáljuk, és elérhetővé tegyük szélesebb réteg számára – erre az internet egy kiváló eszköz. A technika fejlődésével mára a képernyőn is könnyedén meg tudunk jeleníteni háromdimenziós objektumokat, mint például a földgömböket, amely jobban visszaadja a térbeli kapcsolatokat, összefüggéseket.

Témaválasztásomkor a fent említett szempontok mellett szerepet játszott az új, elsősorban webes térinformatikai eszközök megismerésének lehetősége, illetve a már nemzetközi szinten is elismert gyűjtemény, a Virtuális Glóbuszok Múzeumának értékes információkkal és újabb vizualizációs eszközökkel való bővítése. Szakdolgozatom célja létrehozni a digitalizált glóbuszok tárhelyének, a Virtuális Glóbuszok Múzeumban lévő (továbbiakban VGM) egyes földgömbjeihez egy webes felületű vízrajzi névkeresőt, háromdimenziós megjelenítővel. Ebben a keresőben összevethetők egymással a mai és régi óceán- és tengernevek, ezáltal nyomon követhetővé válik a tengeri névrajz alakulása és fejlődése az elmúlt kétszáz évben.

Szakdolgozatomat három nagy részre bontottam. Először a magyarországi glóbuszkészítés történetét, és a VGM-et mutatom be röviden; majd a weboldalhoz szükséges adatbázis felépítéséről írok; ezt követően pedig a weboldal elkészítését ismertetem.

2. Glóbuszok Magyarországon

2.1. A magyarországi glóbuszkészítés történelmi áttekintése

A XIX. századra már a Föld elenyésző területe maradt felfedezetlen, ebből kifolyólag kialakult egy általános igény a feltérképezett területek szemléletesére. Az 1777-es Ratio Educationis előírta a földgömbök iskolákban való alkalmazását, aminek hatására először a tanítók készítettek kéziratos gömböket. 1840-ben Nagy Károly egy 12 hüvelyk átmérőjű (kb. 32 cm) mintagömböt készített angol és francia térképek segítségével. Ez volt az első nyomtatott magyar nyelvű földgömb.

A kiegyezés után az Eötvös József-féle népoktatási törvény értelmében kötelezővé vált a 6–12 éves gyermekek iskolába íratása. Ennek okán megnőtt az igény a földgömbök készítésére iskolai célú használatra. Mivel a kiegyezés után senki sem állított elő magyar földgömböket sorozatgyártásban, Magyarországra kezdetben a prágai székhelyű Felkl cég szállította gömbjeit, amelyeknek névrajzát Gönczy Pál magyarította. Gönczy kortársa Hunfalvy János tervezett egy 32 cm átmérőjű gömböt, amelynek címe *Földünk a legújabb felfedezések nyomán, magyarúl szerkeszté Hunfalvy János*. A cím igazolja, hogy ez tulajdonképpen egy fordítás volt. Egy 1908as közoktatásügyi rendelet kitiltotta az oktatásból azokat a glóbuszokat, amelyeknek tartalmuk elavult. Erre a sorsa jutottak Gönczy és Hunfalvy glóbuszai is.

Ezek pótlására Kogutowitz Manó vállalkozott. A Kogutowitz és társa (1901-től Földrajzi Intézet Rt.) 1897-ben kiadott 25,5 és 51 cm átmérőjű gömböket. Olyannyira igényes munkák voltak ezek a gömbök, hogy külföldi iskoláknak a cég német, olasz és szlovén nyelvű glóbuszokat is szállított.

Turner István csaknem negyven évvel később új glóbuszok készítésébe kezdett, amelyek szerkesztésében részt vett Kogutowitz Károly (K. Manó fia). A glóbuszokat a Magyar Királyi Állami Térképészetben nyomtatták, a névírást Takács József tervezte. Több méretben is készítettek földgömböket, köztük 11, 17, 20, 25, 32 cm-es nagyságban. Politikai kiadása egy 32 cm átmérőjű gömb 1937-ben jelent meg. Ez egy teljesen úgy névrajzzal került forgalomba Kókai Lajos könyvkiadójából.

A második világháborút követően a földgömböket a Turner István és Füsi Lajos szerkesztő munkatárs készítette a Földrajzi Neon Vállalat nevében. Turner cége az államosítás után került a Földrajzi Neon Vállalat alá. A térképkészítő osztály (Fővárosi Neon Földrajzi Tanszerosztálya) átkerült a Kartográfia Vállalathoz. Innentől kezdve a rendszerváltásig Kartográfiai Vállalat szállította a gömböket (Klinghammer 2017).

Ma Magyarországon a Belma Kft az egyetlen glóbuszkészítésre specializálódott cég.

2.2. Virtuális földgömbök

A virtuális földgömbök olyan modellek, amelyek a digitális térben háromdimenzióban jelenítik meg Földünket vagy más égitesteket. Megjelenítésükhöz – mivel háromdimenziós modellek – elengedhetetlen a megjelenítésért felelős hardverek fejlődése, valamint a megjelenítő szoftverek létrejötte. Mára ezek a feltételek egyre inkább elérhetőek a nagyközönség számára is, így kialakult egy igény a glóbuszok digitalizálására.

A virtuális gömbök bár nem tudják helyettesíteni az igazi, kézben tartott földgömbök szerepét, viselkedésük nagyban visszaadja azt ("virtuálisan kézbevehetőek"). Nemkülönben, mint a valóságos földgömbnél, a virtuális gömböt is tudjuk forgatni, sőt egyes részleteket meg tudunk vizsgálni közelebbről, és távolabbról is (a nagyítás funkcióval). Elterjedésükhöz hozzájárult, hogy előállítási költségük jelentékeny mértékben olcsóbb, mint valós társaiké, mindemellett a "nem piacképes" példányok (tematikus gömbök) is helyet kapnak. Talán az egyik legismertebb virtuális földgömb a Google által kifejlesztett Google Earth. A Google Earth (eredetileg EarthViewer3D) 2004 óta van fent a világhálón, legújabb verziója 2017. január 17-én került kiadásra. (Kiss Bettina 2014, Ungvári Zsuzsanna 2009).

2.3. Virtuális Glóbuszok Múzeuma

A régi kézzel készül glóbuszok megóvása és közkincsé tétele érdekében 2007-ben Márton Mátyás és Gede Mátyás létrehozott egy weboldalt, a Virtuális Glóbuszok Múzeumját. "A virtuális tárlatnak az elsődleges célja a Magyarországon készült, vagy valamilyen szempontból magyar vonatkozású föld- és éggömbök bemutatása" (Márton et al 2008). A weboldal a *www.vgm.elte.hu* címen érhető el. A Glóbuszok menüre kattintva megjelenik az ezidáig elkészített és feltöltött virtuális glóbuszok listája. A lista megadott feltételek szerint szűkíthető (pl. átmérő, kiadó). A listából kiválasztott glóbuszra kattintva megjelenik az adott földgömb háromdimenziós modellje. Itt baloldalon látjuk a földgömb makettjét, amit kedvünkre forgathatunk és nagyíthatunk. A jobb oldal pedig tartalmazza az adott glóbusz részletes leírását. Meg kell említenem, hogy ugyanitt lehetőségünk van a földgömb textúráját tartalmazó *kmz* kiterjesztésű fájl letöltésére is a *Google Earth file letöltése* sorra kattintva.



1. ábra: Az egyik glóbuszmodell adatlapja a VGM honlapján

Az oldalon továbbá lehetőségünk van a szerkesztők kapcsolódó publikációit elolvasni a *Publikációk* menü alatt.

3. A felhasznált adatbázis felépítése

3.1. A dolgozatom előzménye: Szabó Virág diplomamunkája

A dolgozatom voltaképpen egy, a tanszéken megírt diplomamunka folytatása, avagy tovább álmodása. Szabó Virág. Ahogyan Virág írja a *Tengeri területek térképi magyar névanyagának fejlődéstörténeti vizsgálata XIX és XX századi földgömbök alapján* 2010-ben leadott diplomamunkája végén: "*reményeim szerint hozzájárulhatok ahhoz, hogy a dolgozat a jövőben kiinduló ponttul szolgálhat a témában bővebben foglalkozni vágyó kutatók számára.*" Virág gondos munkájának folytán létrejött harminc glóbusz vízrajzi neveiről (úgymint óceánok, tengerek, szorosok, fokok) egy összefoglaló táblázat. Virág létrehozott egy külön Excel táblázatot a tengeráramlások neveihez is, ezt az anyagot a dolgozatom elkészítéséhez nem használtam fel, mert csak a glóbuszok egy részén (17) van ilyen név.

Virág minden egyes glóbuszról elkészített egy külön táblázatot, majd ezeket egy nagy, összegző táblázatban egyesítette. A táblázat felépítése a következőképpen néz ki: minden oszlop egy-egy földgömb névrajzát tartalmazza. Ez alól az első oszlop kivétel, itt szerepelnek a tengerek ma használatos nevei. Virág létrehozott egy szabályt, amely a tengernevek felvételének sorrendjét hivatott rendszerezni. E szabályban nem találtam kivetni valót, így átörökítettem az én szakdolgozatomba is. A rendszer lényege, hogy 5 nagy egységet vett alapul, ezek a Csendes-óceán, Atlanti-óceán, Indiai-óceán, a Jeges-tenger és a Déli-Sarki-óceán. A nagy egységeket

külön-külön vizsgálva logikus sorrendben vette fel a hozzájuk tartozó neveket. Mivel feladatom elkészítéséhez egyszerűbb volt a glóbuszokat külön-külön vizsgálni, nem az összefoglaló táblázatot használtam, hanem mindegyik glóbuszhoz önállóan elkészült Excel fájlt (Szabó 2010).

3.2. Shapefile

Első lépésként szükségem volt egy a világ tengereit összefoglaló, azoknak a határait, koordinátáit tartalmazó fájlra. Ezt a *http://www.marineregions.org/downloads.php#iho* oldalról sikerült letölteni. A Marine Regions weboldal célja, hogy létrejöjjön egy egységes, nemzetközileg elfogadott adatbázis, amely pontosan tartalmazza a tengerek határait és az egység földrajzi nevét angolul. A fájlt tehát jelenlegi hivatalos tengeri beosztást tartalmazza.

A letöltött fájl shapefile formátumú. A shapefile főként a geoinformatikai rendszerekben elterjedt vektorgrafikus formátum, amely háromféle geometria alakzatot tartalmazhat: pont, vonallánc, poligon. A tengerhatárok poligonok formájában jelennek meg. A shapefile-okat számos szoftverben megnyithatjuk, én egy ingyenes asztali térinformatikai szoftvert, a QGIS-t választottam.

A letöltött shapefile jó kiindulási alapként szolgált munkámhoz, azonban ahhoz, hogy a készítendő weblapon megjelenítsem, át kellett alakítanom. Többek között a tartalmát is megváltoztattam, ugyanis a tengerbeosztás helyenként kissé mesterkélten hat (pl. Atlanti-óceán északi és déli részre bontva), vagy tartalmaz olyan tengerneveket, amelyeket a gyakorlatban, vagyis a térképeken nem használunk. Akadt néhány olyan tenger volt a shapefile-ban, amelyhez a glóbuszok egyikén sem volt hozzárendelhető földrajzi név. Valószínűleg annak okán, hogy a régmúltban ezek a tengerek még nem léteztek külön egységként. Ezen tengerek poligonjait beleolvasztottam egy nagyobb földrajzi egységbe.

Az összevont poligonok az alábbiak, itt az eredeti angol nevekkel szerepelnek (az 1. számú melléklet tartalmazza a végleges állományban is szereplő nevek magyar megfelelőit):

- Atlantic ocean lett az Inner Seas off the West Coast of Scotland, az North Atlantic Ocean és a South Atlantic Ocean-ből.
- Red Sea lett Gulf of Suez, Gulf of Aqaba és Red Sea egyesítéséből.
- Arab Sea lett Laccadive Sea és Arab Sea-ből.
- Egyesítettem a Malacca Strait-et és Singapore Strait-et.
- Inland Sea és a Japan Sea egyesült.
- Pacific Ocean lett a The Coastal Waters of Southeast Alaska and British Columbia, a South Pacific Ocean, és a North Pacific Ocean-ből

- Mediterrean Sea lett az Alboran Sea, a Balearic Sea, Mediterrean Sea- Western Basin és a Mediterrean Sea- Eastern Basin-ből.
- Sawu Sea az Indian Ocean-nel egyesült.

Végeredményben ezekkel az összevonásokkal 88 poligon jött létre.

Az összevonásokat a következőképpen valósítottam meg:

Miután behívtam a shapefile-t a QGIS-be, először megnyitottam az adott réteg (layer) attribútum tábláját (Open Attribute Table), ahol a poligonokat látjuk felsorolva. Itt kiválasztjuk azokat poligonokat, amelyek össze szeretnénk olvasztani.



2. ábra: Poligonok összevonása QGIS-ben



3. ábra: Az összeolvasztott Vörös-tenger, Szuezi- és Akabai-öböl

A kiválasztás után engedélyeztem a szerkesztést (Toggle Editing), majd a haladó szerkesztések között (Advanced Digitizing) Merge Selected Features ikonra előugrik egy ablak, ahol

beállíthatjuk az új, összeolvasztott poligon attribútumait. Az attribútum tábla egyébként három oszlopban tartalmaz adatot minden egyes poligonról. A poligon, ebben az esetben tenger, vagy óceán angol nevét, Gazetteer számát (az eredeti rendszer szerinti azonosító), illetve egy egyedi a shapefileban tárolt azonosítót. A munkám során ezekből csak az angol nevet hasznosítottam. Új névnek általában annak a poligonnak a nevét állítottam be, amelybe a többi poligon beleolvadt; vagy előfordult, hogy teljesen új nevet kapott pl. az Atlantic Ocean esetében.

3.3. A tengerneveket tartalmazó tábla előállítása

A shapefileből a QGIS program segítségével kimentettem egy CSV formátumú táblázatos fájlt. Ezt a Layer menüre, majd azon belül a Save as opcióra kattintva tudjuk megtenni. A megjelenő ablakban a Format résznél a Comma Separated Value-t (CSV) állítjuk be.



4. ábra: CSV fájl mentése QGIS-ben

Ebben a 88 soros táblázatban szerepeltek a tengerek hivatalos angol megnevezései. Ehhez kellett hozzárendelnem két oszlopot, amely közül az egyik tartalmazza ezen tengerek neveinek a magyar megfelelőit a másik oszlopban pedig egyedi azonosító számot (ID) kapott az összes tenger. Az angol nevek magyar megfelelőit a Cartographia Világatlasz (Cartographia Kft. 1997) segítségével sikerült megadni. Előfordultak olyan esetek, amelynél a földrajzi atlasz nem nyújtott kellő segítséget a magyar név megadásakor, mivel az adott tenger – kicsinysége miatt- már nem volt feltüntetve. Ezen eseteknél más forrásból kellett tájékozódnom.

- Northwestern Passages: Kanadai-szigettenger (Márton Mátyás 2003)
- Bali Sea: Bali-tenger (Dutkó András, 2004)

	А	В	С	D
1	magyar	angol	ID	ID_sajat
2	Ádeni-öböl	Gulf of Aden	38	1
3	Adriai-tenger	Adriatic Sea	28g	2
4	Alaszkai-öböl	Gulf of Alaska	58	3

5. ábra: A tengerneveket tartalmazó Excel fájl felépítése

3.4. A glóbuszokon előforduló neveket tartalmazó tábla létrehozása

Következő lépésként egy új Excel táblát kellett létrehoznom, amely később feltölthető lesz az adatbázisba. Ebben Virág anyagait összegeztem: a különböző glóbuszok vízrajzi neveit tartalmazó táblázatokból egy különálló saját táblázatot hoztam létre, amelynek célja, hogy az egyes glóbuszokon szereplő korabeli vízrajzi neveknek megmutassam a mai megfelelőjét, ha nincs konkrét egyezés, akkor megállapítsam melyik "magasabb rangú" tengerhez tartozik.

A saját Excel létrehozásánál a fokok neveivel nem foglalkoztam: ugyanis ezek a földrajzi alakulatok inkább pontszerűek (egy koordinátapárral leírhatók), semmint felületek. A későbbiekben a kereső akár ezzel is továbbfejleszthető vagy kibővíthető. Virág táblázatából kimásolva a számomra lényeges sorokat egy közel 2700 soros Excel táblázatot kaptam, amelynek minden egyes sorához tartozó földrajzi nevet leellenőriztem a megfelelő glóbuszon. A vizsgálat során többször találkoztam olyan elemekkel, amelyeket nem találtam meg az adott glóbuszon, ezért ezeket a sorokat töröltem. A glóbuszokat vizsgálva ugyanakkor bizonyos eseteknél előfordult, hogy a gömbön feltűnő egy-egy név nem szerepelt a táblázatban, ugyanakkor ugyanez a név egy másik glóbusz táblázatában már felvételre került. Úgy gondolom ezek a nevek a véletlen folytán maradtak ki, úgyhogy ezekkel a lemaradt nevekkel kiegészítettem az adattáblámat. Előfordult, hogy bár véleményem szerint Virág nagyon törekedett a névrajzot pontosan úgy felvinni az Excel fájlba, ahogy az a gömbön szerepel (kisbetű, nagybetű, régies írásmód, kötőjelek), mégis akadtak elírások. Ilyen elírásokat az adott földgömb alapján javítottam. Miután a "gömbi" neveket ellenőriztem, meg kellett adnom mindegyik vízrajzi elemnek a mai megfelelőjét. A legtöbb esetben a tengernév ugyanaz volt, esetleg régies helyesírással írva. A többi esetben leginkább a földrajzi elhelyezkedés segített besorolni a neveket. A mai tengernevet, az előző fejezetben leírtak miatt, elégséges volt egy számmal jelölni (ID_sajat).

Kihívás volt még az öblök szorosok és csatornák nagyobb egységbe való sorolása. Ezzel kapcsolatban csekély mennyiségű szakirodalmat találtam. Ahol tudtam ott Márton Mátyás: 2003-as doktori értekezését, továbbá Dutkó András 2004-es A Világóceán földrajzinév-tára

PhD értekezését használtam. A szerzeményekben hierarchikusan fel van sorolva, hogy mely tengerek mely óceánok részei, illetve néhány szoros, csatorna és öböl is be van osztva nagyobb egységbe. A fel nem sorolt öblök és szorosok és csatornák esetén is próbáltam Márton Mátyás hierarchikus rendjét figyelembe venni. Ezt egy példán keresztül mutatom be.

Ha megnézzük a mai tengereket és határaikat prezentáló shapefile-t, látjuk, hogy *Szunda tenger* helyén a *Jáva-tenger* terül el, az *India Világ Óceán* pedig csak helyesírásban változik e tekintetben. A fenti esetben a kérdés, hogy a *Szunda szorost* mely tengerhez soroljuk be. A Jáva(Szunda)-tenger és az Indiai-óceán a két választási opciónk, mivelhogy ezt a két vízelemet köti össze a szoros. Ha a névadásból kifolyólag logikusan gondolkozva a *Szunda-szorost* a *Szunda tenger* részének tekintjük, aztán összevetjük Márton Mátyás táblázatával, akkor láthatjuk, hogy a Jáva(Szunda)-tenger a Csendes-óceánhoz van besorolva. Tehát feltételezve a logikai kapcsolat a *Szunda-szoros* és a *Szunda-tenger* közt, semmiképpen sem lehet a Csendes-óceánnal egyenrangú India-óceán része. Legtöbb esetben a hierarchiát úgy alkalmaztam, hogy a szorost próbáltam a logikailag hozzá tartozóhoz sorolni. Logikai kapcsolat alatt értem a névbeni egyezést. Amennyiben ez nem teljesült, akkor viszont figyelembe vettem a kettő kérdéses vízfelületnek a földrajzi elhelyezkedését mégpedig úgy, hogy próbáltam a kisebb egységhez csatolni, hogy a későbbiekben könnyebben meg lehessen találni a keresett földrajzi tájat. A különböző glóbuszokon feltűnő ugyanazon szorosokat, csatornákat és öblöket jelölő neveket következetesen mindig ugyanahhoz a tengerhez csatoltam.



6. ábra: Térképrészlet a Szunda-tengerrel

Egy másik érdekes eset volt, amikor több glóbuszon is a feltüntetett tengernév egy ma is létező tenger neve, de egy másik tenger területére van helyezve. A Sulu-tengert más néven nevezik Mindorói-tengernek is. (Wikipedia, Sulu-tenger). Ezeken a bizonyos glóbuszokon a mai Sulu-

tengert Mindorói-tengernek nevezik. Ezzel párhuzamosan viszont feltűnik a Sulu-tenger (legtöbbször Szului tenger helyesírással) név is. Ezek ellenben a mai Celebesz-tenger területét jelölik meg. Ilyesfajta esetekben a régies névhez azt a mai tengernevet (annak azonosító számát) társítottam, amelynek területén a régies nevű tenger fekszik nem pedig a névben vele megegyezőt. Ezt a jelenséget feltüntettem a *megjegyzes* oszlopban.

Az Excel tábla a következőképpen épül fel. A hat oszlopból álló fájl első oszlopában minden sornak egyedi azonosítót hoztam létre. Ez a *rekord_ID* elnevezésű oszlop egy 1-től 2657-ig terjedő számsor. A következő oszlop tartalmazza azt a VGM-ben megkapott gömb azonosítót, amely azonosító egyértelműen meghatározza azt a glóbuszt, amelyről a vízrajz származik. Ezt *gomb_ID*-nek neveztem el. A harmadik oszlop a t*enger_ID* nevet kapta. Ez az a szám, amelyet az előző Excelben hoztam létre az egyes tengerek egyedi azonosítására. Itt kellett mérlegelnem, hogy ha nem létezik egyértelmű névegyezés, akkor melyik tengerhez tartozzon az adott vízfelület. Negyedik oszlopunk a *gomb_nevek* elnevezést viseli. Ez az oszlop tartalmazza a Virág Exceljeiből kimásolt névrajzi anyagot. Az ötödik és hatodik oszlop a felhasznált régies neveket tartalmazó Excel neveit, illetve a megjegyzéseket tartalmazza, ha vannak. Ez a két oszlop elsősorban a saját munkám egyszerűsítése érdekében jött létre, de érdemesnek találtam megtartani, mivel segíthet egy esetleges adatbázis-bővítés esetén.

	А	В	С	D	E	F
1	rekord_ID	gomb_ID	tenger_ID	gomb_nevek	felhasznalt_excel	megjegyzes
2	1	37	7	ATLANTI VILAG TENGER	hunfalvy1	
3	17	37	59	Rio de la Plata	hunfalvy1	én egészítettem ki
4	210	56	68	Mexikoi öb.	gonczy2	Mekikóiról Mexikóira javítva

7. ábra: A glóbuszokon előforduló neveket tartalmazó tábla felépítése

3.5. A gömbök leíró adatait tartalmazó táblázat

A harmadik Excel fájl a VGM oldaláról letöltött a gömbök adatait, információt tartalmazó CSV dokumentum volt. Ebben az esetben egy olyan táblázattal kellett dolgoznom, amely tartalmazta az összes, jelenleg a VGM adatbázisában szereplő, gömb (143 db) adatait. Ezekhez az adatokhoz 23 attribútum tartozik (átmérő, kiadó, szerző, kiadás éve, KML fájl neve stb.). Ebből a táblázatból kellett kiszűrni a számomra értékes és fontos információkat. Legelső lépésben kitöröltem azoknak a glóbuszoknak a tudnivalóit, amelyek nem szerepelnek a szakdolgozatomban. Ez jócskán csökkentette a táblázat méretét. Az oszlopok vizsgálata után a 23 oszlopból 13-at töröltem.

4. A weboldal elkészítése

A névrajazi kereső és megjelenítő felületet egy weboldal formájában képzeltem el. A végleges helye az ELTE Térképtudományi és Geoinformatikai Tanszék Mercator szerverére került, de a fejlesztést és tesztelést a saját gépemen végeztem.

4.1. USBWebserver

Az oldalam teszteléséhez "üzembe helyezése" előtt az USBWebserver nevű programot használtam. Az USBWebserver kombinációja a népszerű webserver szoftvereknek, mint például: Apache, MySQL, Php és PhpMyAdmin. Az USBWebserverrel lehetséges fejleszteni és megjeleníteni a php oldalakat bárhol és bármikor offline állapotban is. Előnye, hogy használható usb-ről és cd-ről egyaránt, vagyis hordozható. A programnak nincsenek járulékos költségei, ingyenesen letölthető. Úgy gondolom, hogy mielőtt feltöltenénk egy oldalt az internetre az **USBWebserver** tökéletes alkalmazás az oldal tesztelésére. (www.usbwebserver.net). A letöltött zip fájlt ki kell csomagolnunk, majd az exe fájllal el is indíthatjuk. Innentől a megjelenő ablak bezárásáig tudunk internet hozzáférés nélkül dolgozni weboldalunkon. A letöltött program legfontosabb részre a root mappa. A localhost elindításakor a program, alapértelmezett esetben, a root mappában jelenlevő index nevezetű fájlt tölti be a böngészőben. A keresősávba a http://localhost:8080/index.php beírásával ugyanez az oldal töltődik be



8. ábra: Az USBWebserver felülete és mappaszerkezete

4.2. phpMyAdmin

A phpMyAdmin egy nyílt forrású eszköz, amit PHP-ban írtak a MySQL menedzselésére az egy webes kezelőfelületen keresztül. Segítségével képesek vagyunk adatbázisokat, táblákat, rekordokat kezelni létrehozni, módosítani, törölni és lekérdezni (Wikipedia, phpMyAdmin) egy grafikus felületen. A MySQL az egyik legelterjedtebb adatbázis-kezelő rendszer. Az SQL, azaz Structured Query Language pedig relációsadatbázis-kezelők szabványos lekérdezési nyelve.

4.3. Az adatbázis létrehozása és feltöltése

Az előző fejezetben bemutatott Excel fájlokat a phpMyAdmin felületén töltöttem fel a MySQLbe. Cél, hogy az egyes adatbázisok között kialakítsak egy relációt. Ha ez sikerült, akkor lekérdezés esetén minden kért és kívánt információt megkapunk.

Összesen három adattáblát hoztam létre importálással:

A táblázatok felépítésükben megegyeznek az előző fejezetben bemutatottal, azzal a kivétellel, hogy az oszlopok neveit kissé megváltoztattam. Mivel a legtöbb programnyelv karakterkészlete angol, az oszlopneveket is ezen elv mentén neveztem, lehagyva a különleges magyar karaktereket. Az oszlopnevek az adott tábla nevének kezdőbetűjével kezdődnek, ezt követően egy alsóvonal után jön oszlop elnevezése. Erre azért volt szükség, hogy a későbbieknek az oszlopokkal való munka során egyszerűbben és érthetőbben tudjak rájuk hivatkozni, megkülönböztetni őket egymástól.

t_magyar	t_angol	t_tenger_id	t_id
Ádeni-öböl	Gulf of Aden	38	1

9. ábra: Példa egy rekordra a tenger táblában

Így létrejött **tenger** néven adatbázis, amely tartalmazza a tengerek magyar és angol neveit, illetve shapefileból kiexportált azonosítót (ezt egyébként nem használtam fel a későbbiekben) és egy egyedi azonosítót.

Úgyszintén az általam készített Excel fájlból hoztam létre a nevek adattáblát.

n_id	n_gomb_id	n_t_id	n_nevek	n_excel	megj		
1	37	7	ATLANTI VILAG TENGER	hunfalvy1			
10. ábra: A nevek tábla egy rekordja							

Harmadik adattáblában (gomb) a VGM-ben fellelhető információk sorakoznak.

g_gomb_id	g_szerkeszto	g_kiadas_varos	g_kiadas_orszag	g_kiadas_ev	g_kmz	g_kiado	g_gomb_atmero	g_gomb_meretarany	g_kep
4	Kartográfiai Vállalat	Budapest	Magyarország	1965	cart2566.kmz	Kartográfiai Vállalat	25	5000000	Carto_1966_25_pol.jpg

11. ábra: Egy rekord felépítése a gomb táblában

A *g_szerkeszto*, *g_kiadas_varos*, *g_kiadas_orszag_ g_kiadas_ev*, *g_kiado* mezőnevek beszédesek, így érthetők, a *g_gomb_id* az adott glóbusz VGM kapott azonosítóját tartalmazza. A *g_kmz* oszlop azokat a *kmz* kiterjesztésű fájlok neveit foglalja magába, amelyek az adott glóbusz textúráját tartalmazzák. a *g_gomb_atmero* oszlopban lévő szám az adott fölgömb átmérője centiméterben megadva. A *g_gomb_meretarany* mező a földgolyó méretarányát (csak méretarányszám) jelenti. Az utolsó oszlop a *g_kep* amely a Virtuális Glóbuszok Múzeumában megjelenő kis indexképnek a fájlneve.

Az adattáblák közti reláció ismertetését egy ábrán keresztül szeretném felvázolni.



12. ábra: Az adattáblák közti reláció szemléltetése

Az ábrán jól megfigyelhető, hogy a *gömb* (az ábrán gomb) és a *tenger* tábla táblákat a *nevek* táblával tudjuk összekötni. A *nevek* tábla n_gomb_id lesz az idegen kulcs, amely a *gömb* tábla g_gomb_id elsődleges kulcsára mutat, valamint a *nevek* tábla n_t_id is szintén egy idegen kulcs, amely a *tenger* tábla a t_id mezőre (elsődleges kulcs) mutat. Ezt a relációt MySQL-ben lekérdezéssel hoztam létre. Ez az SQL lekérdezésem a következőképpen néz ki:

SELECT * FROM NEVEK, TENGER, GOMB WHERE n_gomb_id=g_gomb_id AND n_t_id=t_id

A lekérdezés ebben az állapotában listát készít az összes elérhető információról (SELECT *), vagyis melyik korabeli tengernév, melyik gömbön található és annak mi az idegen és magyar nyelvű, mai megfelelője.

Az adattáblákat létrehozásuk után a phpMyAdmin felületén keresztül kiexportáltam, megkapva a három sql fájlt, amelyeket a weboldal üzembe helyezésénél (Mercator szerverre való telepítésénél) használtam fel.

4.4. A weboldal "külleme"

A weboldal kinézete és elrendezése nemcsak esztétikai kérdés: biztosítja a könnyű áttekinthetőséget és segíti a felhasználót az információk megértésében. A szakdolgozatom lényegi részét ugyan nem érinti ez, mégis úgy éreztem, hogy az igényes megjelenés minőséget kölcsönöz neki.

Mivel a Virtuális Glóbuszok Múzeumához tartalmilag szorosan összefűzhető weboldalról van szó, úgy gondoltam, hogy kinézetben is összefüggőnek kell lennie. Ezért az első prototípus egy a VGM-hez hasonló oldal lett. Mivelhogy a VGM oldalán a menüsáv sok lehetőséget rejt és az én oldalam esetében erre nem volt szükség, egy oldalon megjeleníthető minden fontos információ, a menüpontokat és ezzel együtt a több oldalas oldal tervét elvetettem.

Így jött létre a második, a VGM-hez már kevésbé hasonlító honlap. Célom volt, egy átlátható, minimalista, könnyen használható oldalt felépíteni, ezért a menüsáv csupán egy keresősávot és a Gömb azonosítókat tartalmazó lenyíló kapcsolót tartalmaz. Három részt tudunk megkülönböztetni ebben a menüsávban. Az első ahová a keresési mintánkat írjuk, a második, ahol kiválasztjuk, hogy milyen adattáblában keressen, illetve harmadik a keresőgomb, aminek megnyomására aktiválódik a keresésünk. A fejléc tartalmazza az oldal címét, illetve jobb oldalában négy sort (A weboldal készítésének idejét, illetve készítőjét, a VGM-re mutató linket, a tanszék honlapjára mutató linket, és a tanszék koordinátáit tartalmazó Google Maps térképrészletre mutató linket. A linkek mindegyike kattintásra új ablakban jelenik meg.). A menüsáv alatt a képernyőt két egységre bontottam. A bal oldali egység tartalmazza a keresés eredményét tartalmazó táblázatot, a jobb oldali egység pedig a kiválasztott földgömb háromdimenziós modelljét. A weboldal a Google Chrome böngészőjéhez lett optimalizálva.

4.5. A CSS formázás

A weboldal a VGM-től eltérően más színeket, más betűtípust használ. Ezen felül tartalmaz olyan részeket, mint például az árnyékolás, "hover" effektek vagy ikonok, amelyeket a VGM nélkülöz. A honlap CSS kódja, amely tartalmazza a honlap kinézetét egy külön mappában foglal helyet. Ugyancsak a CSS fájlban formáztam meg a PHP-kóddal létrehozott táblázatokat. Ahhoz, hogy a honlapon megjelenjenek a CSS-ben írt változtatások, hivatkoznunk kell rá az főoldal fejlécében, jelen esetben az index.php fájlban. Ezt a következőképpen tudjuk elérni

<link rel="stylesheet" href="css/style.css"/>

A *link* létrehozza a kapcsolatot a dokumentum és a külső forrás között, a *rel* meghatározza a kapcsolat jellegét a hivatkozott dokumentum és a jelenlegi dokumentum között a *href* pedig a hivatkozott dokumentum helyét adja meg.

					© 2017 Beszkid Arthur	
	(à Manaini	n árdrana a			& Virtuális Glóbuszok Műz	euma 2.0
	🦉 vizrajzi	nevkereso			in ELTE Térképtudományi e	és Geoinformatikai Tanszék
					✓ 1117 Budapest, Pázmány	Péter sétány 1/A
	at	Név	γ Q Gõ	mb azonosítók		
	Keresett szöveg: at					a
	Régi név	Gömb	Magyar név	Angol név		
0	ATLANTI-ÓCEÁN	Kartográfiai Vállalat, 1965	Atlanti-óceán	Atlantic Ocean	and the second second	
•	La Plata	Kartográfiai Vállalat, 1965	La Plata	Rio de La Plata		
0	Szent Mátyás-öböl	Kartográfiai Vállalat, 1965	Atlanti-óceán	Atlantic Ocean	A State State State State	
•	Drake-átjáró	Kartográfiai Vállalat, 1965	Atlanti-óceán	Atlantic Ocean	A BRITHE STATISTICS	
0	Mocambique-csatoma	Kartográfiai Vállalat, 1965	Mozambiki- csatorna	Mozambique Channel	The star was a first	
۲	ATLANTI-ÓCZEÁN	Kogutowicz Manó, 1905	Atlanti-óceán	Atlantic Ocean		
0	La Plata	Kooutowicz Manó, 1905	La Plata	Rio de La Plata		
•	St.Mátyás-öb.	Kogutowicz Manó, 1905	Atlanti-óceán	Atlantic Ocean	and the second sec	
0	Katteg.	Kogutowicz Manó, 1905	Kattegat	Kattegat	Cause The Contract	
•	Tatár-öb.	Kogutowicz Manó, 1905	Japán-tenger	Japan Sea		K 7
	ΑΤΙ ΑΝΤΙ ΟΟΕΑΝ	222 1011	Atlanti-óceán	Atlantic Ocean		

13. ábra: A honlap kinézete a táblázattal

4.6. A megjelenítő

Mielőtt röviden ismertetném a HTML kódot, szükségesnek látom, hogy az oldal háromdimenziós megjelenítőjével foglalkozzam. Számos háromdimenziós megjelenítő létezik, amelyet glóbuszok megjelenítéséhez használunk. A VGM a WebGL technológiát használja az X3DOM keretrendszerrel. A WebGL egy "web alapú grafikus könyvtár", amely a JavaScript programozási nyelvet támogatja plugin mentes háromdimenziós grafikai lehetőségekkel (Wikipedia,WebGl, khronos.org) Én az úgyszintén WebGL technológát alkalmazó megjelenítőt, a *Cesiumot* használtam fel.



14. ábra: Cesium megjelenése alaphelyzetben

A Cesiumot az oldal jobb oldali részében helyeztem el. А beágyazáshoz le kellett tölteni a programot a hivatalos weboldalról, átmásolni a könyvtárunkba, majd importálni a HTML dokumentum fejlécében. A Cesium számos előre definiált jellemzővel bír, amelyek közül jó néhányra nem volt szükségünk. Az alap megjelenítésnél a háttérben látszanak a csillagok, a

Nap és a Hold, illetve a Föld atmoszférája is. A lehetőségek között van egy idővonal is, ahol az

időt tudjuk állítani, tekerni. Ezt az időbeni változások szemléltetésére lehet felhasználni. Az *Cesium* kiegészítése még a jobb felső sarokban található öt kis ikon (kereső, *HOME* gomb, nézet váltó, műhold felvétel megjelenítő, súgó). Ezek közül kizárólag csak a *HOME* gombot hagytam meg, amelyre kattintva a földgömb nézet alaphelyzetbe áll vissza. Az egyes funkciók blokkolásához a *false* értéket kellett megadni a kódban. Alább a környezet optimalizálása olvasható (légkör, égbolt: Nap, Hold és csillagok megjelenítésének letiltása):

```
viewer.scene.skyAtmosphere.show = false;
viewer.scene.skyBox.show = false;
viewer.scene.sun.show = false;
viewer.scene.moon.show = false;
```

Mivel a fölgömbünk háttér textúráját is kivettük, meg kell adnunk neki egy színt. Erre a szimpla feketét találtam az oldal stílusába leginkább illőnek.

viewer.scene.backgroundColor = Cesium.Color.BLACK;

4.6.1. Glóbusz textúrák

A különböző glóbuszok textúráit *kmz* kiterjesztésű fájlok tárolják. Célom, hogy ezeket a textúrákat a gömb felszínére rávetítsem. Azt, hogy ezeket a *KMZ* fájlokat honnan töltse be *Cesium* két lehetőségem volt. Az egyik, hogy a glóbuszok *KMZ* fájljait letöltve, a honlappal megegyező szerveren, egy mappában tároljuk őket, és innen hívjuk őket elő. A másik, hogy a VGM oldaláról töltjük be őket. Az előbbinek előnye, hogy mivel nem kell mindig letölteni a fájlt, ezért gyorsabban töltődik be a textúra, az utóbbi előnye pedig, hogy egy mindig naprakész *KMZ* textúrát tud betölteni, ezenfelül nem kell kétszer tárolni egy nagyméretű mappát. Egy új glóbusz felvételekor sem kell külön foglalkozni a textúrával, hiszen az adattábla bővítése után a kód automatikusan be tudja tölteni azt.

Alaphelyzetben, az oldal megnyitásakor a *gonczy3.kmz* (az egyik Gönczy földgömb) fájl töltődik be. A különböző földgömbök textúrái között lehet váltani is, erre később visszatérek.



15. ábra: A honlap a Gönczy-féle földgömb textúrájával

4.6.2. A tengerhatárokat tartalmazó fájl átalakítása a webes keresőhöz

Annak érdekében, hogy a megjelenítőn szemléltetni lehessen az egyes tengerek kiterjedéseit, létre kellett hoznom egy olyan réteget a földgömbhöz, amelyen szerepelnek az általam használt tengerek poligonjai. A létrehozott réteg a glóbuszon a VGM-ből letöltött textúrával együttesen lesz jelen. A feladat elkészítéséhez szükség lesz egy, a világ tengereit ábrázoló *KML* kiterjesztésű fájlra. Azért ezt a formátumot választottam, mert egyrészt könnyen létrehozható térinformatikai szoftverekből, másrészt a *Cesium* is fel tudja dolgozni. Az általam összevont, tengerhatárokat tartalmazó shapefileból exportáltam ki a *KML* fájlt. Mivel ez a shapefile túl részletesen ábrázolja a partvonalat, tehát rengeteg töréspontot tartalmaz, mérete is nagy lesz. Emiatt a weboldalba való betöltődése lassú, ezért szükséges volt a partvonalat egyszerűsíteni.



16. ábra: Görögország partvonala az egyszerűsítés előtt és után

Térinformatikai szoftverben generalizált határrajzot készíteni "kézi" rajzzal igen hosszadalmas, ezért automatizáláshoz folyamodtam. Több térinformatikai szoftver vagy alkalmazás is tartalmaz vonalegyszerűsítési algoritmusokat. A QGIS-ben a lehetőségeink korlátozottak, ugyanis olyan eszközre lesz szükség, amely képes poligonokat egyszerűsíteni. Így két eszköz jöhetett volna számításba, közülük az egyik a SimpliPy, de két korábbi szakdolgozatban is

bebizonyították, hogy a működése nem mindig helyes (Kocsis 2015 és Balogh 2015). A másik modul, a PolygonSimplifierben lévő algoritmusok eredménye nem adott "szép" képet.

Ezért elsőként a kereskedelemben is kapható szoftverrel, az ArcGIS-szel próbálkoztam, de a generalizáló algoritmus lefutása után több hibát jelzett, amelyet nem sikerült megoldani (pl. az egyszerűsítést nem végezte el minden elemen). Végezetül az ArcGIS szoftvert elvetettem és egy online felületet használtam fel, ez a mapshaper.org oldal.

Az oldal felépítése egyszerű, könnyen használható (17. ábra). Legelőször is be kell importálni a shapefile *SHP* és *DBF* kiterjesztésű fájljait. Ezután a jobb sarokban található *Simplify* gombra kattintva, az előugró ablakon beállíthatjuk az egyszerűsítés paramétereit. Fontos, hogy a *Prevent shape removal* beállítás mellett pipa legyen, hiszen ekkor a program biztosit minket, hogy az egyszerűsítés következtében biztosan nem vesztjük el a legkisebb poligonunkat sem. Módszernek a Visvalingam-féle egyszerűsítési algoritmust választottam.

Az eljárás lényege, hogy fokozatosan eltávolítsa a pontokat a lehető legkisebb "alakbeli" változással. Annak a megállapításához, hogy melyik pontot távolítsa el, kiszámol az összes ponthoz egy effektív területet. Egy pont effektív területe a vele szomszédos pontokkal alkotott

háromszög területe. A kezdő- és végpont nem rendelkezik két szomszédos ponttal, tehát az algoritmus hozzájuk nem rendel effektív értéket, ezáltal nem is törli őket a folyamat során. A legkisebb területű háromszöggel, effektív azaz értékkel rendelkező pontot törli a vonalból, ezt háromszögek követően а területét újraszámolja, majd az eljárást megismétli mindaddig, amíg csak a két végpontunk marad, ezért meg kell határoznunk egy toleranciaértéket, amely következtében az algoritmus addig fut ameddig a legkisebb



17. ábra: Visvalingam módszerének vizualizációja

effektív terület is nagyobb lesz, mint a megadott toleranciaérték (Visvalingam és Whyatt, 1993). Az 16. ábra esetében a vonalon a kék háromszög a legkisebb területű, tehát balról a negyedik pontot törli az algrotimus. A kettős szomszédos háromszög területét ezután újraszámolja.

Miután a csúszkán megadtuk az egyszerűsítés értékét százalékban, nincs más teendőnk, mint az export gombbal (jobb felső sarok) shapefile formátumban kimenteni az egyszerűsített fájlunkat.



18. ábra: Partvonal egyszerűsítés a mapshaper.org oldalon

Utolsó lépésként ezt a shapefilet kell *KML* formátumban elmenteni. A QGIS Save as opciójával próbálkoztam legelőször, de a kapott *KML* fájlt megnyitva, nem a várt felépítést kaptam. A kapott *KML* fájl nem tartalmazta a poligonok neveit (tengernevek) és egy több elemet összefogó CDATA mezőt sem (SchemaData-val dolgozott, amit a *Cesium* nem értelmezett). Ezért a Global Mapper programmal próbálkoztam. Itt a shapefile megnyitása után az Export Vector Format menüre rákattintva majd a *KMZ/KML* fájl típust kiválasztva tudtam kimenteni a *KML* kiterjesztésű fájlomat.



19. ábra: KML fáfjl kimentése shapefileból Global Mapper segítségével

Ezt a *KML* fájlt (továbbiakban tengerhatarok.kml) hasonlóan töltjük be a *Cesium*-ba, mint a glóbuszok textúráit, annyi kivétellel, hogy itt csak egy fájlt, egyszer kell betöltenünk. Ezt JavaScripttel tettem meg. Ugyanitt lehet beállítani a fájlra vonatkozó formázásokat is, ahol megadtam egy 1000 méteres magasságot. hogy a poligonunk véletlenül se metssze a gömb makettet. A poligonokra egy alapbeállítás érvényes, miszerint minden poligonnak van körvonala. Ezt feleslegesnek tartottam, ezért "false" értéket állítottam be az outline-nak.

A *tengerhatarok.kml* réteg elsődleges feladata, hogy kiemelje a poligont, vagyis a tengert, amelyen éppen a kurzor van, illetve kattintásra információkat közöljön róla.

A poligon kiemelést a legkönnyebben úgy lehet elérni, hogy adunk egy alapszínt a poligonoknak, majd egy olyan színt, amire akkor vált át, hogyha a kurzor éppen felette van. Ez a két szín nálam ugyanaz a kék szín. A különbség csupán annyi, hogy alapszínnek van 10%-os átlátszóságot adtam meg, a változó színnek pedig egy 25%-os átlátszóságot. Így szerintem szépen kiemeli őket, de nem rontja a gömb textúrájának olvashatóságát, továbbá a határok extrém generalizáltsága sem zavaró.

```
var mouseOverColor=Cesium.Color.BLUE.withAlpha(0.25);
var alapColor=Cesium.Color.BLUE.withAlpha(0.1);
```

A *Cesium* beépített funkciója, hogy a poligonra kattintva előugrik az ún. InfoBox amely a *KML* fájlban az adott poligon CDATA elemeit tartalmazzák. Nálam a CDATA tartalmazza az egyes poligonok neveinek angol megfelelőit. A CDATA-ban szereplő tag-ekat egyszerű HTML szabvánnyal lehet formázni.

4.6.3. A KML fájl készítésénél felmerült nehézségek

Ha jobban megfigyeljük a shapefile-t láthatjuk, hogy az eredetileg gömbön értelmezett poligonok vannak két dimenzióba átalakítva. Ezt úgy oldotta meg a program, hogy a 180. hosszúsági foknál elmetszette és "kiterítette" a poligonokat. Sajnos ezzel egy olyan helyzet állt elő, hogy a három tenger (Bering-tenger, Csukcs-tenger, Csendes-óceán) is két poligonként szerepel a gömbökön. Ezzel együtt a megjelenésük is olyan, mintha az egyik oldaluk a 180-as hosszúsági fok volna (lenti ábra). Ennek kiküszöbölésére szükséges egy pillantást vetni a *KML* fájl rájuk vonatkozó részeire.

<coordinates>

141.8537000000,48.7571454545,0

</coordinates>

Egy poligon minden egyes pontja a fentiekhez hasonlóan néz ki. Három részből áll, amiket vesszők határolnak: a földrajzi hosszúság, földrajzi szélesség, és a magasság méterben megadva. A megoldás, hogy a két poligont úgy kell egyesíteni, hogy az egyik poligon

koordinátái átlógjanak a dátumválasztón. Ehhez a dátumválasztó egyik oldalán helyet foglaló poligon koordinátáit át kell számolni. Én a pozitív hosszúságú koordinátákkal rendelkező poligont hagytam meg, ehhez számoltam át a dátumválasztó másik oldalán lévő negatív hosszúságú koordinátákat. Ezt úgy kell megtenni, hogy a 180°-hoz hozzáadjuk a koordináta dátumválasztótól vett távolságát. Tehát -173,5° esetében a kapott eredmény 186,5°lesz.

Egy poligon rengeteg koordinátával rendelkezik, ezért a feladat felgyorsítása és egyszerűsítése érdekében átszámolásra egy programot volt célszerű írni. Ezt témavezetőm javaslatára PHPben írtam meg. Előzetesen kimásoltam az egyik tenger koordinátáit egy text fájlba. A program megnyitja ezt a fájlt, soronként beolvassa, majd a vesszőnél elvágja a sorokat, így egy három oszlopos tömb jön létre. A tömb első elemét (földrajzi hosszúság) átszámolja a fentebb említett módon, majd ugyanolyan formátumban, ahogy a text-ben volt kiírja. Ezt az eredményt kell manuális a *KML* fájlba beilleszteni a megfelelő helyre. Ennek eredményeképpen megkapjuk az összeolvasztott poligonunkat.



20. ábra: Bering-tenger egyesítés előtt és után

Ezzel a módszerrel illesztettem össze a Csendes-óceán ketté szelt poligonjait is. Viszont ebben az esetben a *Cesium* nem tudta rendesen megjeleníteni az összeolvasztott poligont. Valószínűleg a poligon olyan sok töréspontot tartalmaz, amelyet a *Cesium* már nem tudott feldolgozni (a Google Earth-ben nem volt probléma a megjelenítéssel). Ezért a Csendes-óceán poligonját külön, a Google Earth-en rajzoltam meg. A létrehozott poligont *KML* fájlként kimentettem, majd a Csendes-óceán koordinátáit lecseréltem az új poligon koordinátáira az eredeti dokumentumban.

Amikor a textúrákat teszteltem, figyelmes lettem, hogy a Kartográfia Vállalat 1962-es gömbnek a textúrája a *Cesiumon*, a tenger határait tartalmazó poligonhoz képest, el van tolódva nyugatra. Ennek oka, hogy ennél a földgömbnél még a kezdőmeridián a ferroi volt. Később ez a VGMben javítva lett. Ugyancsak a textúrák tesztelésénél vettem észre, hogy egyes földgömbök textúrájának egy része nem töltődik be (21. ábra). Mivel a textúrák az VGM-en keresztül töltődnek le, gyanítottam, hogy a letöltési kapcsolatban lehet baj. Átírva a kódot, egy lokális helyről töltötte be a *KMZ* fájlokat, de a hiba még mindig fennállt. Meglátásom szerint a hiba a *Cesium* megjelenítőben keresendő, amit egyelőre nem tudtunk megoldani (Kíséletet az említett a textúrát alkotó képek méretének csökkentésére (2048×2048 pixel) is, de ez sem segített.)



21. ábra: A hiányos textúra

4.7. A HTML-kód

A HTML (HyperText Markup Language) egy leíró nyelv, amelyet weboldalak készítéséhez fejlesztettek, és mára internetes szabvánnyá vált. Legújabb változata a HTML5.

Az oldal HTML kódját az index.php fájl tartalmazza. Ezek a vezérlő szerkezetek adják meg az oldal felépítését. Ehhez fűztem hozzá a style.css fájlt, amely a HTML alapot megformázza. Ahhoz, hogy a php kódot értelmezni tudja, kell a webserver és hogy a fájl kiterjesztése .php legyen. A kód emellett a <?php ... ?> közé kerül. A HTML dokumentum egyetlen részletét, a keresősávot szeretném ismertetni.

```
<form method="GET">
	<input class="search" type="text" placeholder="Keresési minta..
"id="minta" name="minta" required autocomplete="off" />
		<select name="mezo" id="mezo">
		<select name="mezo" id="mezo">
		<option value="n_nevek">Név</option>
		<option value="n_nevek">Név</option>
		<option value="g_gomb_id">Gömb ID</option>
		</select>
		<button id="gomb" type="button" onclick="loadDoc()" ><i
class="fa fa-search" aria-hidden="true"></i></button>
</form>
```

Vízrajzi névker	reső	
Keresési minta	Név Q Gömb ID	— 1. opció — 2. opció
"minta"	"mezo"	opoio

22. ábra: A keresősáv felépítése

A "minta" névre keresztelt részbe lehet beírni a keresett szöveget, a "mezo" részben pedig kiválasztani a feltételt a két opció (Név vagy Gömb ID) közül.

A kódban a button részhez hozzárendeltem egy *onclick* függvényt. Az *onclick* függvény az a függvény, amelyik akkor hívódik meg, amikor a gombra rákattintanak. A gomb a "Keresés" felirat helyett, egy nagyító képét tartalmazza. A függvényt loadDocnak neveztem. Szerepe, hogy a keresés végrehajtása az egész oldal újratöltődése nélkül következzen be, ezáltal megelőzve azt, hogy a *Cesium* valamint a *Cesiumon* megjelenő kétféle textúra minden egyes keresésnél újra betöltődjön. Amellett, hogy az adatforgalmat lényeges csökkenti, az oldalt gyorsabbá, gördülékenyebbé teszi, ezáltal biztosit egy magasabb fokú böngészési élményt. Erre AJAX-ot (Asynchronous JavaScript And XML) használtam. Az AJAX használatával képesek vagyunk a háttérben információt cserélni a web szerverrel. Az AJAX a böngészőkbe beépített XMLHttpRequestnek és a JavaScriptnek a kombinációja. Az XMLHttpRequest felelős a webserverrel való kommunikációért, a JavaScripttel pedig a kapott információt jelenítjük meg. function loadDoc() {

xhttp=new ActiveXObject("Microsoft.XMLHTTP"); }

Létrehozzuk a XMLHttpRequest objektumot xhttp változóként. Ahhoz, hogy az összes böngészőben működjön a funkció, először ellenőrizzük, hogy támogatja e böngészőnk a XMLHttpRequest objektumot. Ha nem akkor ActiveXObject-et hozunk létre (régi Explorer böngészők). A modern böngészők már kivétel nélkül támogatják az XMLHttpRequest-et. Ez fogja hordozni az információt a szerver és a honlap között.

```
xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        var valasz=document.getElementById("left_column");
        valasz.innerHTML = xhttp.responseText;}
```

Az onreadystatechange függvény szerint, ha a változónk megérkezett és válaszra kész (readyState==4), akkor a válasz az általunk megjelölt div-be fog betöltődni.

A keresőbe bevitt adatokhoz létrehoztam két változót (amely tartalmazza a mezo és a minta értékét), majd az open paranccsal megnyitottam a GET metódusú kérést. Ez jelen esetben a php kódunkat tartalmazó kereso.php fájl címsorába kerül az fenti két változó értéke, onnan olvassa ki őket a PHP-kód. A send paranccsal elküldtem a kérést.

4.8. A PHP-kód

A PHP (Hypertext Preprocessor) egy széles körben használt nyílt forráskódú általánost célú programnyelv, amely különösen alkalmas a webes fejlesztéshez és beágyazható a HTML-be. A PHP kód a szerveroldalon hajtódik végre, aminek következtében HTML kódot generálhatunk. Ez lesz elküldve a kliensoldali felhasználónak, tehát a felhasználó nem fogja látni a mi PHP kódunkat. Ez különbözteti meg például a JavaScript programozási nyelvtől.

A php-kód a <?php ?> tag-ek közé esik. Ez hozza létre a kapcsolatot a honlap és MySQL adatbázis közt. A kód egy külön fájlban foglal helyet, amely az index.php-val azonos mappában van. A következőkben a kód felépítéséről és fontosabb részleteiről lesz szó.

A kódunk első két sora új változónevet ad meg a keresőbe bevitt értékeknek. A *mint* és *mez* neveket először a *loadDoc* függvényben deklaráltam.

```
$szoveg=$_GET["mint"];
```

\$lista=\$_GET["mez"];

A kódunk legelőször megvizsgálja, hogy megadtunk-e valamilyen értéket a keresősávban

if (isset(\$szoveg)&&\$szoveg!='') {

Amennyiben igen, akkor a beírt szöveget kiírja a program.

print "Keresett szöveg: ".\$szoveg."";

A keresés a MySQL-ben létrehozott adattáblákban történik. Az elkészített adattáblák végül a mercator.elte.hu szerveren lettek elhelyezve, ahhoz, hogy ezekben keressünk, először csatlakozni kell az adatbázis-szerverhez:

\$db=mysqli_connect('host','felhasználónév','jelszó','adatbázis');

Aztán a csatlakozást ellenőrizzük. Ha nem jön létre kapcsolat azt írassuk ki a hibát.

if (mysqli_connect_errno())

die("Nem sikerült csatlakozni: ".mysqli_connect_error());

Következő lépésben végrehajtjuk a lekérdezésünket

\$eredmeny=mysqli_query(\$db, "select * from nevek, tenger, gomb where {\$lista} like '%{\$szoveg}%' and n_gomb_id=g_gomb_id and n_t_id=t_id"); A lekérdezésben összekapcsoltam a táblákat. A \$*lista* változóval azt adjuk meg, hogy melyik adattábla melyik oszlopában keressük a megadott szöveget. A '%{\$szoveg}%' pedig a megadott keresendő szöveg, ahol a % azt engedi meg, hogy a beírt részlet előtt és után lehessenek még karakterek. Ezzel a kódrészlettel lekérünk minden adatot a szerverünkről, amely a megadott kritériumoknak eleget tesz, majd ezt az *eredmeny* változóba helyezzük.

Ezután meg kell vizsgálni, hogy a megadott szempontok alapján a létrehozott *eredmeny* változónk tartalmaz-e elemeket. Hasonlóan annak vizsgálatához, hogy írtunk-e valamit a keresőbe, itt is 'if-else' elágazást kell alkalmazni.

if (mysqli_num_rows(\$eredmeny)>0)

Azt, hogy van e eredményünk a program úgy állapítja meg, hogy összeszámolja az eredmény változónkban tárolt sorokat, és amennyiben ez egyenlő nullával, akkor nincs eredményünk. Ebben az esetben a következő részlet lép életbe.

else

```
print "<b>Nincs találat! Próbálja újra!</b>";
```

Ellenkező esetben az eredményeket egy táblázatban jelenítettem meg. Először a táblázat fejlécét hozzuk létre, ahol már létrehozzuk az eredmények kiírására az oszlopokat.

Öt oszlopot hoztam létre a fenti kód szerint. A tag-ek határolják a táblázat fejlécének celláit, a a sorokat. Az első oszlop fejlécébe nem került cím, mivel ez az oszlop egy úgynevezett radio button-nek ad helyett, amelyről később még visszatérek.

A megjelenítéshez le is kell töltenünk az lekérdezett információkat. Erre a *mysqli_fetch_assoc* függvényt használjuk. Ez az eredményünket asszociatív tömbként adja vissza. Ezen "végigfutva" egy táblázatba töltjük be az adatokat.

\$sor=mysqli fetch assoc(\$eredmeny)

A print parancesal pedig kiíratjuk a lekért információkat.

```
print "
<input type='radio' name='globusz'
value='http://terkeptar.elte.hu/vgm/kmz/".$sor['g_kmz']."'
onclick='kmzValtas();'>

value='''.$sor['r_magyar']."
''.$sor['r_magyar']."
''.$sor['r_magyar']."
''.$sor['r_magyar']."

value=''.$sor['t_magyar']."
''.$sor['r_magyar']."
''.$sor['r_magyar']."
```

A lekért információkat ugyanúgy, mint a fejlécet cellákra bontottam (). Öt cellát, azaz oszlopot, különböztetünk meg. Minden cellában a kívánt információt töltjük be az adattábláinkból, úgy, hogy az információt tartalmazó adattábla megfelelő mezőjére hivatkozunk. (pl.:'t_magyar')

Az első oszlopban létrehoztam egy radio buttont amelyet, ha kiválasztunk, akkor egy az index.php állományunkban megírt, JavaScript függvény hajtódik végre (*kmzValtas()*).

```
function kmzValtas() {
```

```
var radios=document.getElementsByName('globusz');
for (i = 0 ; i < radios.length; i++) {
    if (radios[i].checked) {
      globe_url=radios[i].value;
      kmz.entities.removeAll();
      kmz.load(globe_url).then(function () {
        var e=kmz.entities.values;});
break;}}</pre>
```

A funkció lényege, hogy a háromdimenziós megjelenítő textúráját lecseréli annak a gömbnek a textúrájára, amihez a keresett vízrajzi név tartozik. A térképtár oldalán az összes *kmz* fájlnak van egyedi url címe, amelynek az utolsó tagja maga a *KMZ* fájl neve. Például: *http://terkeptar.elte.hu/vgm/kmz/gonczy3.kmz*. Ezt a *kmz* fájlnevet, amely az url utolsó tagja, tartalmazza a gömb adattáblánk g_*kmz* oszlopa. A radio button értékének pedig ezt az url címet adjuk meg, úgy, hogy utolsó tagja mindig a megfelelő glóbusz *kmz* fájljára mutasson. A *kmzValtas* függvényünk pedig megkeresi, letölti, majd megjeleníti ezt a gömbünkön.

A második oszlopban a tengerek nevei szerepelnek, amely tulajdonképpen egy gomb, amelyre kattintva végrehajtódik az a függvény, amellyel a háromdimenziós földgömb az adott tengerhez "fordul (*zoomTo()*).

function zoomTo(nev) {

viewer.flyTo(elemek[nev]);

viewer.selectedEntity=elemek[nev];}

Működése abban áll, hogy megvizsgálja az adott sorhoz tartozó t_magyar oszlop értékét, majd, ha talál ennek megfelelőt a tengerhatárokat tartalmazó *KML* fájlban a poligonok nevei közt, akkor kamera képe ráközelít az adott poligonra. Az oszlop tartalma, ami a gombra is felkerül, az egyik opcionálisan kereshető tartalom, a glóbuszokon fellelhető névrajz.

A harmadik oszlopban találhatóak a legfontosabb információk a glóbuszról, a szerkesztő neve és a kiadás ideje. Ezentúl ez oszlop tartalma egyben linként szolgál, amely az adott földgömb VGM oldalára irányit át bennünket.

A negyedik és ötödik oszlop a vízrajz mai nevét magyar, illetve angol formában adja meg.

Ezeket az információkat megjelenítő táblázatot a CSS fájlban formáztam meg.

A PHP kódunk végén pedig szükséges lezárnunk a kapcsolatot:

mysqli_close(\$db);}

4.9. Weboldal elérhetősége

A weboldal a Mercator tanszéki szerveren kapott végleges helyet. Az adattáblákat tartalmazó sql fájlok és az oldalt tartalmazó és kiegészítő fájlok és mappák kerültek fel ide. Az oldal elérhető a http://mercator.elte.hu/~jfc8ms címen ezenfelül a VGM weboldaláról is.

5. Összegzés

Dolgozatomban a magyar glóbuszok tengeri neveinek összehasonlításához hoztam létre egy webes felületet. A feladat során a weboldal elkészítéséhez szükséges lépéseket mutattam be. A dolgozat megvalósításához több térinformatikai szoftvert és programozási nyelvet is kellett használnom.

Az első részben rövid áttekintést adtam a magyarországi glóbuszkészítés történelméről, a virtuális földgömbökről és ezek legnagyobb magyarországi lelőhelyéről, a Virtuális Glóbuszok Múzeumáról. Ezen kívül bemutattam Szabó Virág diplomamunkáját, amely szakdolgozatom alapjául szolgált.

A második részben részletes bemutatásra került az adatbázis elkészítésének folyamata, bemutatva, hogyan egyeztettem össze a régi tengerneveket a mai megfelelőjükkel. Szintén bemutatásra került a létrehozott adattáblák közti logikai kapcsolat.

Ezután foglalkoztam a honlap felépítésével, működésével és kapcsolatával a MySQL adatbázissal. Külön tárgyaltam a megjelenítéskor felmerült hibákat és ezen problémák megoldásait. Ismertettem a weboldal működéséhez szükséges kódrészleteket.

Munkám célja, egy felület létrehozása, ahol a tengernevek változásait tudjuk megfigyelni a magyar glóbuszokon az idő változásával.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Ungvári Zsuzsannának, aki hasznos és értékes tanácsaival, illetve ötleteivel rengeteg segítséget nyújtott a téma kiválasztásában és a szakdolgozat elkészítésében.

Irodalomjegyzék

Balogh Dániel, 2015: A térképi generalizálás automatizálásának vizsgálata a vízrajz példáján

Szakdolgozat, ELTE, Budapest, 2015.

Cartographia, 1997: Világatlasz.

Cartographia Kft, Budapest, 1997.

Dutkó András Ákos, 2004: Az óceánfenék földrajzinév-tára és elektronikus atlasza.

PhD értekezés, ELTE Budapest, 2004.

Kiss Bettina, 2014: Kéziratos glóbusz digitalizálása és újraalkotása modern módszerekkel.

Szakdolgozat, ELTE, Budapest, 2014.

Klinghammer István, 1998: A föld és éggömbök története.

Eötvös Kiadó, Budapest, 1998.

Klinghammer István, 2017: A föld- és éggömb - az oktatás eszköze, a lakás dísze.

Magyar tudomány, 2017. (178. évf.) 3. sz. 377-380. old.

Kocsis Viktória, 2015: A térképi generalizálás automatizálásának bemutatása a partvonalak és az országhatárok példáján keresztül.

Szakdolgozat, ELTE, Budapest, 2015.

Márton Mátyás, 2003: A Világtenger (A térképezéstől a komplex leírásig).

MTA Doktori értekezés, Budapest, 2003.

Márton Mátyás, 2011: Új Hunfalvy-Felkl földgömb.

Geodézia és kartográfia, 2011. (63. évf.) 2. sz. 10-12. old.

Márton Mátyás, 2012: A Világtenger kartográfus szemmel.

ELTE, Budapest, 2012.

Márton M.–Gede M.–Zentai L., 2008: Föld- (és ég-) gömbök 3D-s előállítása - Virtuális Földgömbök Múzeuma és digitális virtuális restaurálás.

Geodézia és kartográfia, 2008. (60. évf.) 1-2. sz. 36-42. old.

Szabó Virág 2010: Tengeri vetületek térképi magyar névanyagának fejlődéstörténetei vizsgálata a XIX. és XX. századi földgömbök alapján.

Diplomamunka, ELTE, Budapest, 2010.

Ungvári Zsuzsanna, 2009: Földgömbtérképek készítése digitális vetületi transzformációval.

Szakdolgozat, ELTE, Budapest, 2009.

M. Visvalingam–J. D. Whyatt, 1993: Linegeneralisation by repeated elimination of points.

The Cartographic Journal, 1993. 30 (1). 46-51. oldal.

Elektronikus források

AJAX

https://www.w3schools.com/xml/ajax_php.asp

Cesium

http://cesiumjs.org/

CSS

https://www.w3schools.com/css/

Hivatalos tengeri beosztást tartalmazó shapefile

http://marineregions.org/downloads.php

HTML

https://www.w3schools.com/html/

MySQL

https://www.mysql.com/

PHP

https://www.php.net

https://www.w3schools.com/php/

phpMyAdmin

https://www.phpmyadmin.net/

Szulu-tenger

https://hu.wikipedia.org/wiki/Sulu-tenger

USBWebserver letöltési helye

http://www.usbwebserver.net/en/

Virtuális Glóbuszok Múzeuma

http://terkeptar.elte.hu/vgm/2/?lang=hu

Visvalingam-algoritmus ábra forrása

http://www.cs.gettysburg.edu/~ilinkin/courses/Spring-2016/cs107/assignments/a6-files/visvali-whyatt.png

WebGL

https://www.khronos.org/webgl/

https://hu.wikipedia.org/wiki/WebGL

A honlapok utolsó elérése 2017. május 15.

Mellékletek

1. számú melléklet

Magyar	Angol	Magyar	Angol
Ádeni-öböl	Gulf of Aden	Kattegat	Kattegat
Adriai-tenger	Adriatic Sea	Kelet-kínai-tenger	Eastern China Sea
Alaszkai-öböl	Gulf of Alaska	Kelet-szibériai-tenger	East Siberian Sea
Andamán-tenger	Andaman or Burma Sea	Kelta-tenger	Celtic Sea
Arab-tenger	Arabian Sea	Korall-tenger	Coral Sea
Arafura-tenger	Arafura Sea	La Manche	English Channel
Atlanti-óceán	Atlantic Ocean	La Plata	Rio de La Plata
Azovi-tenger	Sea of Azov	Labrador-tenger	Labrador Sea
Baffin-öböl	Baffin Bay	Laptyev-tenger	Laptev Sea
Bali-tenger	Bali Sea	Ligur-tenger	Ligurian Sea
Balti-tenger	Baltic Sea	Japán-tenger	Japan Sea
Banda-tenger	Banda Sea	Jáva-tenger	Java Sea
Barents-tenger	Barentsz Sea	Jeges-tenger	Arctic Ocean
Bass-szoros	Bass Strait	Jón-tenger	Ionian Sea
Beaufort-tenger	Beaufort Sea	Kaliforniai-öböl	Gulf of California
Bengáli-öböl	Bay of Bengal	Kanadai-szigettenger	Northwestern Passages
Bering-tenger	Bering Sea	Kara-tenger	Kara Sea
Bismarck-tenger	Bismarck Sea	Karib (Antilla)-tenger	Caribbean Sea
Bone-öböl	Gulf of Boni	Lincoln-tenger	Lincoln Sea
Botteni-öböl	Gulf of Bothnia	Makasari-szoros	Makassar Strait
Bristoli-csatorna	Bristol Channel	Malaka-szoros	Malacca Strait
Celebesz-tenger	Celebes Sea	Maluku-tenger	Molukka Sea
Csendes-óceán	Pacific Ocean	Márvány-tenger	Sea of Marmara
Csukcs-tenger	Chukchi Sea	Mexikói-öböl	Gulf of Mexico
Davis-szoros	Davis Strait	Mozambiki-csatorna	Mozambique Channel
Déli-sarki-óceán	Southern Ocean	Nagy-Ausztráliai-öböl	Great Australian Bight
Dél-kínai-tenger	South China Sea	Norvég-tenger	Norwegian Sea
Égei-tenger	Aegean Sea	Ohotszki-tenger	Sea of Okhotsk
Északi-tenger	North Sea	Ománi-öböl	Gulf of Oman
Fehér-tenger	White Sea	Perzsa (Arab)-öböl	Persian Gulf
Fekete-tenger	Black Sea	Rigai-öböl	Gulf of Riga
Filippínó-tenger	Philippine Sea	Salamon-tenger	Solomon Sea
Finn-öböl	Gulf of Finland	Sárga-tenger	Yellow Sea
Flores-tenger	Flores Sea	Seram-tenger	Ceram Sea
Földközi-tenger	Mediterranean Sea	Skaggerak	Skaggerak
Fundy-öböl	Bay of Fundy	Sulu-tenger	Sulu Sea
Gibraltári-szoros	Strait of Gibraltar	Szent Lőrinc-öböl	Gulf of St-Lawrence
Grönlandi-tenger	Greenland Sea	Tasman-tenger	Tasman Sea
Guineai-öböl	Gulf of Guinea	Thai (Sziámi)-öböl	Gulf of Thailand
Halmahera-tenger	Halamahera Sea	Timor-tenger	Timor Sea
Hudson-öböl	Hudson Bay	Tirrén-tenger	Tyrrhenian Sea
Hudson-szoros	Hudson Strait	Tomini-öböl	Gulf of Tomini
Indiai-óceán	Indian Ocean	Vizcayai-öböl	Bay of Biscay
Ír-tenger és Szent György- csatorna	Irish Sea and St. George's Channel	Vörös-tenger	Red Sea

Nyilatkozat

Alulírott, Beszkid Arthur (Neptun azonosító: JFC8MS) nyilatkozom, hogy jelen szakdolgozatom teljes egészében saját, önálló szellemi termékem. A szakdolgozatot sem részben, sem egészében semmilyen más felsőfokú oktatási vagy egyéb intézménybe nem nyújtottam be. A szakdolgozatomban felhasznált, szerzői joggal védett anyagokra vonatkozó engedély a mellékletben megtalálható.

A témavezető által benyújtásra elfogadott szakdolgozat PDF formátumban való elektronikus publikálásához a tanszéki honlapon

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2017. május 15.

.....

a hallgató aláírása