

EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
TERMÉSZETTUDOMÁNYI KAR

## A Soroksári-Duna interaktív vízisport-térképe

### DIPLOMAMUNKA TÉRKÉPÉSZ MESTERSZAK

*Készítette:*

Bakó Tamás

*Témavezető:*

Dr. Gede Mátyás

egyetemi docens

ELTE Térképtudományi és Geoinformatikai Tanszék



Budapest, 2021

**EÖTVÖS LORÁND TUDOMÁNYEGYETEM**  
**INFORMATIKAI KAR**  
**TÉRKÉPTUDOMÁNYI ÉS GEOINFORMATIKAI TANSZÉK**

**DIPLOMAMUNKA TÉMABEJELENTŐ**

Hallgató adatai:

Név: **Bakó Tamás**

Neptun-kód: JKI9Y6

Képzési adatok:

Szak: térképész, mesterképzés (MA/MSc)

Tagozat: Nappali

Belső témavezetővel rendelkezem

Témavezető neve: **Gede Mátyás**

munkahelyének neve: ELTE IK Térképtudományi és Geoinformatikai Tanszék

munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/A.

beosztás és iskolai végzettsége: Egyetemi docens, PhD

A diplomamunka címe:

**A Soroksári-Duna interaktív vízisport-térképe**

A diplomamunka témája:

*(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben diplomamunka témájának leírását)*

Olyan webes és/vagy mobilalkalmazás készítése, mely segíti a vízitúrázókat a Soroksári-Dunán. Az OpenStreetMapről letölthető és a vízügyi hatóságtól igényelt adatok kiegészítése saját felméréssel.

Budapest, 2020.11.25.

# TARTALOMJEGYZÉK

|                                                       |    |
|-------------------------------------------------------|----|
| Interactive watersport map of the Soroksári-Duna..... | 5  |
| Bevezetés.....                                        | 6  |
| A magyar vízisport-térképek .....                     | 7  |
| A mintaterület tulajdonságai.....                     | 8  |
| Adatgyűjtés és -feldolgozás.....                      | 9  |
| Medermélység .....                                    | 9  |
| Kikötőhelyek .....                                    | 11 |
| Parttípusok.....                                      | 12 |
| Vízi növényzet.....                                   | 12 |
| POI.....                                              | 13 |
| Egyéb pontszerű elemek.....                           | 14 |
| Hidak .....                                           | 15 |
| Térinformatikai alkalmazásfejlesztés .....            | 16 |
| A térképobjektum .....                                | 16 |
| Háttértérképek .....                                  | 16 |
| Medermélység WMS-réteg.....                           | 17 |
| GeoJSON rétegek .....                                 | 18 |
| Rétegkezelő .....                                     | 20 |
| Helymeghatározás .....                                | 20 |
| Kezdőnézet .....                                      | 22 |
| Áttekintőtérkép .....                                 | 22 |
| Vízi útvonalterv .....                                | 24 |
| Jelmagyarázat .....                                   | 26 |
| Formázás.....                                         | 27 |
| A felhasználói felület egyéb elemei.....              | 27 |
| Eredmények, további célok .....                       | 28 |

|                                                           |    |
|-----------------------------------------------------------|----|
| Összegzés, konklúzió .....                                | 29 |
| Ábra- és táblázatjegyzék.....                             | 30 |
| Ábrajegyzék.....                                          | 30 |
| Táblázatjegyzék .....                                     | 30 |
| Irodalomjegyzék .....                                     | 31 |
| Köszönetnyilvánítás .....                                 | 32 |
| Mellékletek .....                                         | 33 |
| Engedély .....                                            | 33 |
| Képernyőképek az asztali- és mobil webes felületről ..... | 34 |
| A CD-melléklet fájlrendszere .....                        | 35 |
| Diplomamunka eredetiség nyilatkozat .....                 | 36 |

# INTERACTIVE WATERSPORT MAP OF THE SOROKSÁRI-DUNA

## Abstract

In this study I present the process of creating the Soroksári-Duna web application.

I will expound the preparatory work: data request, digitization, operations with spatial data layers, and on-site measurement. I will describe in detail the steps of web development: the structure of HTML, CSS formatting, and JavaScript. This includes the Leaflet function-library, geolocation function, route planning with watercrafts, overview map links, and layer operations.

As a result a short chapter summarizes the content of this study, and the working website can be accessed at the following link:

 <https://bakotamas.github.io/rsd/>

**Keywords:** *interactive map, watersport-map, Soroksári-Duna, web application, web cartography, web mapping, GIS, JavaScript, Leaflet, Maptiler, Mapserver, geolocation, route planning, canoesport, recreation map*

## BEVEZETÉS

Magyarország bővelkedik sportolásra, víztúrázásra, és szabadidős evezésre alkalmas tavakban, folyókban, víztározókban és holtágakban. Ezek túlnyomó részén a már meglévő infrastruktúra lehetővé is teszi azt: csónakházak és kikötőhelyek találhatóak az ország számos pontján.

Budapest kedvelt rekreációs vízparti területei közé tartozik a Római-part, ahonnan gyorsan elérhető a Szentendrei-Dunaág, az Újpesti-öböl partján fekvő Nép-sziget, valamint a Soroksári-Duna, ami egyesíti az előbb említett két helyszín előnyeit. Szabályozott vízállása, elhanyagolható sodrása (0,2-0,4 km/h) és gyér hajóforgalma kezdők és hobbisportolók számára is nyugodt, biztonságos közeget jelent, ahol lehetőség van a vízi sporteszközök használatának elsajátítására, az alapvető manőverek és rutinmozgás megtanulására. Teljes egészében hajózható, 57,3 kilométeres hosszával pedig akár többnapos túrázásra is alkalmas. Az ideális körülmények mellett amolyan ráadásként a sportoló változatos, szebbnél szebb tájakon evezhet keresztül, miközben a magyar vizes élőhelyekre jellemző, nagy fajgazdagságú, nyüzsgő élővilág veszi körül.

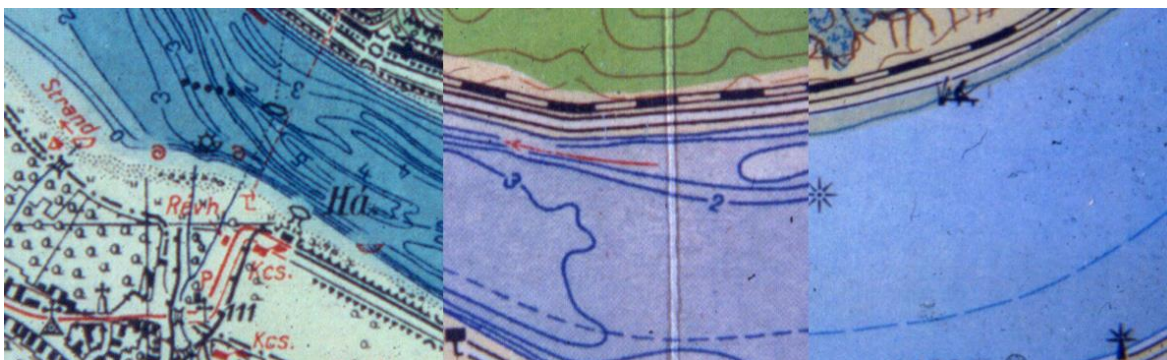
Az ország sportolásra alkalmas vízfelületeiről – így a Soroksári-Dunáról – azonban nincsenek kellően részletes és egyben aktualizált nagyméretarányú térképek, amik segítenék a túratervezést és a terepi tájékozódást is.

Ezt a hiányt hivatott pótolni a Soroksári-Duna interaktív térképe web- és mobilalkalmazás, ami ingyenesen elérhető, valamint a jelenkori elvárásokhoz illeszkedve telefonon is használható.

## A MAGYAR VÍZISPORT-TÉRKÉPEK

A magyar vízisport-térképek történetében megfigyelhető egy – tematikus tartalmat érintő – információsűrűség-csökkenési tendencia. Az 1930-as években a Magyar Királyi Állami Térképészet kiadásában megjelent „angyalos” vízisporttérkép-sorozat köteteiben volt a legszélesebb körű tematika ábrázolva a lehető legnagyobb tartalmi sűrűség mellett. A medermélység szintvonalas megjelenést kapott, felkerültek továbbá a zátonyok, egyéb víz alatti veszélyekre figyelmeztető elemek, a sodorvonal, az ajánlott víziút vonal, és partvonal minőség szerint kategorizálva. Nem feledkeztek meg a vizes határnevekről sem.

Az 1970-es években a Kartográfiai Vállalat vízisport-térképe már nagyobb szintközökkel ábrázolja a mélységvonalakat, eltűnik a partvonal kategorizált megjelenítése, és a zátonyokat sem jelzi külön. A '80-as években megjelent újabb kiadásban csak romlik a helyzet. Ekkor a térképolvasó már sem a partról, sem a mederről nem kap lényegi információt. Az újabb térképek esetében is rendszeresen elhagyják a mélység- és partvonalábrázolást, holott ezek a leginkább sokatmondó térképi elemek egy vízitúrázó számára, amellet, hogy alig adnak hozzá a tartalmi sűrűséghez. (Faragó, 2020)



1. ábra: A Tematikus tartalmat érintő információsűrűség-csökkenési tendencia, balról jobbra: 1930, 1970, 1980

A Soroksári-Duna interaktív vízisport-térképe webtérkép mivoltából adódóan másfajta kartográfiai ábrázolásmóddal, lehetőségekkel operál, mint nyomtatásban megjelent társai, azonban tartalmi szempontból az angyalos-térképek színvonalának elérése a cél.

Ennek alapján a téradatbázis tartalma egy részletes medermélység-térkép, a partszakaszok ábrázolása parttípus szerint kategorizálva, kikötőhelyek és minden egyéb vízi építmény, természetes elem, ami a mederben található, vízijármű kölcsönző helyek és szolgáltatásaik, valamint a céltematikát támogató vízparti elemek.

## A MINTATERÜLET TULAJDONSÁGAI

A mintaterületként kijelölt bő 11 kilométeres folyószakasz a Soroksári-Duna legészakabbra eső része. Folyásirány szerinti kezdőpontja a Rákóczi híd szomszédságában található Kvassay-zsilip. Innen mintegy 10 kilométeren keresztül még Budapest területén folyik dél felé, majd a városhatár átlépése után ezer méterrel az M0 autópályát hidalja át, ami a kijelölt szakasz vége is egyben.

A Kvassay-zsilipet sporteszközzel nem lehet használni, ha valaki így érkezik a Nagy-Duna felől, a zsilip mellett kézzel kell átvinnie azt. Innen dél felé haladva rögtön a bal parton fekszik a vízirendőrség telephelye, ahonnan járőrözni indulnak a motorcsónakokkal.

Megközelítőleg három kilométerrel lejjebb keresztezi a folyót a Gubacsi híd, majd attól másfél kilométerre a víz felett átívelő közművezeték: a Csőhíd. Ezek között helyezkedik el a mintaterület csónakházainak túlnyomó része. Rengeteg egyesület és klub található ezen a rövid szakaszon és nyáron a vízen minden napszakban zajlanak az edzések, és eveznek a túrázók.

A Csőhídtól három kilométerre van a csepeli komp, a BKK  14 jelű járata, ami *Csepel-Királyerdő* és *Soroksár, Molnár-sziget* állomások között közlekedik reggel 7 órától este 8-ig fél óránként minden nap, gyalog és személyautóval egyaránt igénybe vehető. A komp kötélvontatásos rendszerű, keresztezni az útvonalát abban az esetben lehet, ha a még kikötött állapotában fehér zászló van rá felvonva. A közlekedő komp útvonalát még mögötte sem jó ötlet keresztezni, mert a kötélzet ilyenkor a vízfelszín közelébe emelkedik, ami a vízi járművet megrongálhatja, utasát kiboríthatja.

Még a komp és a Csőhíd között fekszik az edzés közben tragikusan elhunyt kétszeres olimpiai bajnok kenus, Kolonics György emlékhelye és egyesülete, a Csepeli Kajak-Kenu Klub. A part mentén a Gubacsi hídtól a kompig halad a róla elnevezett sétány, aminek déli végén szobra is megtalálható.

A komptól délre az M0-s hídig tartó három kilométeres szakaszon már nincsenek csónakházak, a part mellett egy darabig lakóházak és nyaralók sorakoznak, majd az utolsó másfél kilométeren az ág összeszűkül: a Csepeli oldalon hatalmas nádassal borított lápos vizes élőhely terül el, amire közvetlenül a szűkületnél természetvédelmi területet jelölő tábla hívja fel a figyelmet.



## ADATGYŰJTÉS ÉS -FELDOLGOZÁS

A projekt megvalósítása az adatgyűjtéssel kezdődött. A térkép tematikájából adódóan sokféle forrásra volt szükség. A dinamikusan változó elemekhez terepi felméréssel és vízi helyszíneléssel kellett adatokat nyerni, míg az időben állandó, illetve jelentéktelen változékonyságot mutató entitások esetében az adatok többnyire kérvényezés után – vagy már eleve – rendelkezésre álltak, így a megfelelő konverziók elvégzése után a projekt számára feldolgozható formába kerültek.

Különös figyelmet kellett fordítani arra, hogy bizonyos adattípusokba tartozó rétegeket nem elég csupán a mintaterületre ábrázolni. A kontinuum típusú adatokat a projekt teljes területére el kellett készíteni, különben a szakaszhatároknál később problémát jelentene a különálló részek topológiai helyes összefésülése.

Az alábbiakban az egyes rétegek alapjául szolgáló elemek sajátosságai kerülnek bemutatásra.

### *Medermélység*

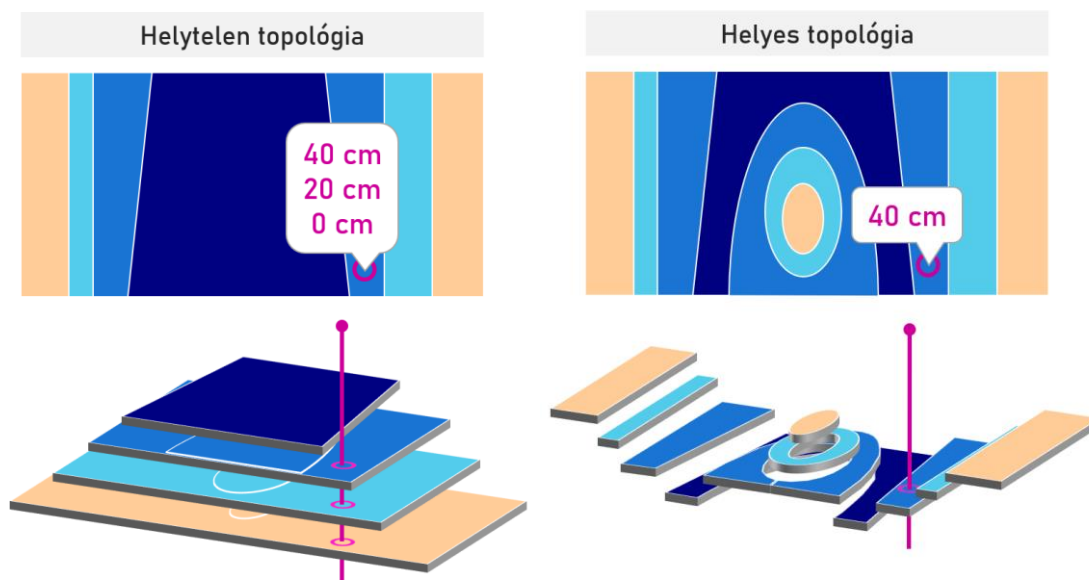
Első és egyik legfontosabb lépése volt a munkának, a részletes szintvonalas vízmélységábrázolás elkészítése. A raszteres állományú medermodell-adatokat az Országos Vízügyi Főigazgatóság Műszaki Téradat Szolgáltató Főosztálya bocsátotta rendelkezésre a Soroksári-Duna-ág teljes hosszáról. Az adatokat geotiff<sup>1</sup> formátumban kaptam, a színezés a tengerszint feletti magasság szerint történt.

Ahhoz, hogy az interaktív térképhez is használható vektoros formában lehessen kezelni a réteget, először a QGIS Feldolgozás eszköztár Raszter átvétel szekciójából a Szintvonal algoritmus segítségével alakítottam át shapefile réteggé, paraméterként 20 cm szintközt megválasztva. A parancs futtatása ~80 percet vett igénybe. Ezek után az elkészült szintvonalas réteget a Vektor geometria eszközök Vonalak felületté algoritmusával poligonréteggé konvertáltam, hogy színkitöltést lehessen adni a szintfelületeknek a térképen.

---

<sup>1</sup> A GeoTIFF képek esetében a .tiff/.tif kiterjesztésű raszteres állományokban metaadatként vannak eltárolva a képhez tartozó földrajzi információk, így azokat térinformatikai rendszerekbe betöltve georeferálás nélkül is helyesen lehet megjeleníteni.

Ezen a ponton azonban egy előre nem várt hiba merült fel. A poligonokat mélység szerint rendezve, egymásra rétegezve, lyukak nélkül generálta a program, ami többféle topológiai hibát is jelentett:



2. ábra: Topológiai hibák kijavítása

A színek különböző mélységtartományokat jelölnek: 0 cm (part) 20 cm 40 cm 60 cm

A hibacsoport olyan esetekben jött létre, amikor az egyenletesen mélyülő mederben kiemelkedések, zátonyok vagy szigetek fordultak elő. Az azonos mélységű szintvonalakat a poligonra alakítás algoritmusa egy tagolatlan, belső gyűrűk nélküli felületté konvertálta. A réteg szimbólumszint-beállításánál fordítva kellett felsorakoztatni a mélységfelületeket, különben a legnagyobb kiterjedésű, ám legsekélyebb vízszintet jelentő poligon az összes többi fedte volna. Ezen körülmények miatt azonban a mélyebb részek takarták el helytelenül a kiemelkedéseket, illetve ez az anomália a mélységlekérdezések során újabb hibához vezetett. Legtöbb esetben ugyanis kettő, vagy még több szintfelület-poligon volt egymásra rétegezve, ezért – ahol egyébként a térképi megjelenés akár helytálló is lehetett – ezekre a helyekre egyszerre több vízmélység is vonatkozott (2. ábra: Topológiai hibák kijavítása), ami nyilvánvalóan lehetetlen.

A hibák orvosolására témavezetőm, Dr. Gede Mátyás a Global Mapper szintfelület-generáló eszközét javasolta, így végre elkészült a topológiaiilag is helyes medermélység shapefile-réteg.

## Kikötőhelyek

Szintén elengedhetetlen a vízi közlekedéshez a kiépített vízreszálló-helyek ismerete. Erre vonatkozóan semmilyen nyilvántartás nem állt rendelkezésre, az OpenStreetMap adatbázisa pedig a közösségi szerkesztés miatt erősen hiányos, tulajdonképpen csak elvétve találhatók ilyen típusú elemek benne. A Google Earth műholdképe alapján ugyan ezek nagyrésze digitalizálható, de a növényzet zavaró hatása, a kategóriákba sorolás, és az aktualizáció miatt elengedhetetlen a szisztematikus terepi felmérés.

A folyóág egy bizonyos szakaszáról rendelkezésre állt Google „Utcakép”, így az megkönnyítette a munkát. Ezt a szakaszt ellenőrzés céljából, valamint a többi szakaszt a felmérés miatt C1-es kenuval jártam végig, és a hajóra szerelt kamera képei alapján (10 másodpercenként automatikus exponálásra volt állítva) kategorizáltam a kikötőhelyeket. A poligonok felvételét a QGIS Haladó digitalizálás eszköztárával végeztem, ahol többek között lehetőség van pontos oldalhosszúságok és szögek megválasztására. A réteg megjelenése típus szerint kategorizált, az attribútumtábláját az alábbiak szerint építettem fel:

1. táblázat: Az RSD\_kikotohelyek.shp fájl attribútumtábla-felépítése

| MEZŐNÉV    | MEZŐÉRTÉK  | LEÍRÁS                                                                                               |
|------------|------------|------------------------------------------------------------------------------------------------------|
| nev        | „szöveg”   | Kikötőhely neve (opcionális)                                                                         |
| típus      | steg       | Stég típusú, mederhez rögzített kikötőhely                                                           |
|            | ponton     | Ponton típusú vízen úszó kikötőhely                                                                  |
|            | molo       | Móló típusú, partra épített kikötőhely                                                               |
|            | allohajo   | Rögzített helyzetű hajó, úszóház típusú kikötőhely                                                   |
|            | magassteg  | Nagyobb motorcsónakok, jachtok számára kialakított kikötőhely, csónakból a ki- és beszállás nehézkes |
|            | rom        | Rossz állapotú, balesetveszélyes, kikötésre nem ajánlott kikötőhely                                  |
| megjegyzes | „szöveg”   | Megjegyzés (opcionális)                                                                              |
| magan      | igaz-hamis | Kikötési tilalmat jelez                                                                              |
| vizitúra   | igaz-hamis | A kikötőhely egyben Vízitúra-megállóhely is                                                          |
| horgasz    | igaz-hamis | Horgászállás, kikötés csak szükség esetén javasolt                                                   |

## Parttípusok

A partvonal ábrázolása vízisport-térképek esetében megköveteli a minőség szerinti kategorizálást, emiatt külön réteggként kell, hogy szerepeljen a térképen. Ez az osztályozás segíti a túrázót a megfelelő kikötő-, megálló- illetve táborozóhely megválasztásában.

2. táblázat: Az RSD\_partszakaszok.shp fájl attribútumtábla-felépítése

| MEZŐNÉV    | MEZŐÉRTÉK  | LEÍRÁS                                                            |
|------------|------------|-------------------------------------------------------------------|
| típus      | lankas     | Lankás part: leginkább alkalmas kikötésre, csónak kihúzására      |
|            | meredek    | Meredek part: nehéz partra jutni, csónakot nem lehet kihúzni      |
|            | alamosott  | Alámosott part: kikötésre alkalmatlan, partra lépni veszélyes     |
|            | felhanyt   | Felhányt part menti kövek: kikötésre alkalmatlan                  |
|            | lejtos     | Lejtős kőfal: a meredekség miatt nehézkes a feljutás              |
|            | fuggoleges | Függőleges kőfal: Kikötni, partra szállni lehetetlen              |
|            | javitott   | Vízszintnél alig magasabb javított part (lehet deszkaburkolat is) |
| megjegyzes | „szöveg”   | Megjegyzés (opcionális)                                           |

## Vízi növényzet

A parttípusok ábrázolása önmagában nem szolgál elegendő információval a vízitúrázók számára, mivel az egyébként kikötésre alkalmas partszakasz és a vízfelület határán vízi járművel átjárhatatlan növényzet is elhelyezkedhet. Ez a Soroksári-Dunán jellemzően nád vagy sás.

A vízi növényzet rétegen felületi elemként látszanak ezek az átjárhatatlan növényzeti elemek, így a kiterjedésük is szembetűnő. Külön kategória jelzi továbbá az olyan hínáros területeket a vízben, ahol sűrű előfordulásuk lassítja, akadályozza a haladást a vízi járműre történő feltorlódással, felakadással.

3. táblázat: Az RSD\_novenyzet.shp fájl attribútumtábla-felépítése

| MEZŐNÉV | MEZŐÉRTÉK | LEÍRÁS                                  |
|---------|-----------|-----------------------------------------|
| típus   | nád       | Áthatolhatatlan vízi növényzet          |
|         | hinar     | A haladást lassító víz alatti növényzet |

A növényzet felmérése a kikötőhelyekhez és a parttípusokhoz hasonlóan történt. Digitalizálás során a part menti növényzet határát a vízén pontosan, a felmérés szerint vettem fel, azonban a poligonokat a partra „kifuttattam”, és csak elnagyoltan zártam le azokat. Ezek után növényzet rétegen lefuttattam a QGIS Metszés algoritmusát, a vízfelületet használva metsző réteggé. Így kialakult a végleges réteg, amin helyesen szerepeltek a poligonok mind térbeli helyzetüket, mind pedig a topológiát tekintve.

## POI<sup>2</sup>

A tematika szempontjából érdekes helyek a POI rétegre kerültek. Ez szolgál a túrázók számára olyan kiegészítő információkkal, amik alapján az útvonalat és a pihenőket megtervezik. Leginkább a szolgáltatások adják ennek a rétegnek az alapját, de az emlékművek és szobrok is helyet kaptak rajta.

4. táblázat: Az RSD\_poi.shp fájl attribútumtábla-felépítése

| MEZŐNÉV    | MEZŐÉRTÉK | LEÍRÁS                             |
|------------|-----------|------------------------------------|
| nev        | „szöveg”  | POI neve                           |
| tipus      | csh       | Csónakház, hajóbérlési lehetőség   |
|            | hotel     | Szálláshely                        |
|            | etterem   | Büfé, étterem                      |
|            | strand    | Strand, fürdő, uszoda              |
|            | vm        | Vízitúra-megállóhely hivatkozással |
|            | szobor    | Emlékmű, szobor                    |
| megjegyzes | „szöveg”  | Megjegyzés (opcionális)            |
| web        | „szöveg”  | Weboldal (opcionális)              |
| nyitva     | „szöveg”  | Nyitvatartás (opcionális)          |
| telefon    | „szöveg”  | Telefonszám (opcionális)           |
| email      | „szöveg”  | E-mail cím (opcionális)            |

<sup>2</sup> Point Of Interest - hasznos, érdekes helyek: Felhasználási terület függő, hogy ez mit jelent. Általában turisztikai látványosság, hotel, étterem, bankautomata, gyógyszertár, orvosi rendelő, üzlet, benzinkút, mozi, iskola, templom stb. Ebben az esetben a csónakházak, vízitúra-megállóhelyek, és egyéb evezéshez kapcsolódó objektumok is a kategóriába tartoznak.

A webes térképeknek egyik nagy előnye, hogy egy bizonyos elemhez tartozó információkat nem kell külön tárgymutató fejezetben elhelyezni. A képernyőn megjelenő egyszerű elemmel interakcióba lépve, akár szövegbuborékban, új ablakban, vagy felugró értesítésben közvetlenül elérhetők az adatai. Az információ lehet nagy terjedelmű, görgethető szöveg is, de hang- és videófájl épp úgy beágyazható.

## Egyéb pontszerű elemek

Ez a réteg magában foglalja mindazokat az egyszerű pontszerű objektumokat, amelyek nem rendelkeznek névvel, további információkkal, így nincs szükség kattintásra felugró ablak megjelenítésére. Ezek közé tartozik a folyamkilométer, a vízi közlekedési táblák és egyéb informatív jelek.

5. táblázat: Az RSD\_alappont.shp fájl attribútumtábla-felépítése

| MEZŐNÉV    | MEZŐÉRTÉK         | LEÍRÁS                                     |
|------------|-------------------|--------------------------------------------|
| nev        | „szöveg”          | Objektum neve (opcionális)                 |
| típus      | fkm               | Folyamkilométer a Tassi-zsiliptől számítva |
|            | limany            | Limányos hely                              |
|            | tablakomp         | Komp tábla                                 |
|            | tablavezetek      | Felsővezeték tábla                         |
|            | tablamegallas     | Kötelező megállás tábla                    |
|            | tablatiloskikotni | Kikötni tilos tábla                        |
| megjegyzes | „szöveg”          | Megjegyzés (opcionális)                    |

A vízi KRESZ-táblák elsősorban a motoros hajókra vonatkoznak, azonban a rekreációs célú, emberi erővel hajtott járművek utasai számára is szolgálnak információval, illetve tájékozási pontként is remekül használhatóak.

Fontos megjegyezni, hogy ahol piros kertes, fehér alapú táblán „SPORT” felirat van átlósan pirossal áthúzva, kedvtelési célú vízijárművel közlekedni tilos.

## Hidak

Nevével ellentétben ez a réteg minden olyan objektumot tartalmaz, ami a Soroksári-Dunát keresztben átíveli. A felhasználó tájékozódását segíti evezés közben a felsővezetékek, hidak helyének ismerete. Erre a rétegre került fel továbbá a komp típus is, ami a hidakhoz hasonlóan átkelési lehetőséget biztosít, így még a vízitúrára érkezéskor lehet hasznos.

6. táblázat: Az RSD\_hidak.shp fájl attribútumtábla-szerkezete

| MEZŐNÉV    | MEZŐÉRTÉK    | LEÍRÁS                             |
|------------|--------------|------------------------------------|
| nev        | „szöveg”     | Átívelő objektum neve (opcionális) |
| típus      | kozlekedes   | Közúti-, vagy vasúti híd           |
|            | komp         | Komp                               |
|            | kozmuvezetek | Közművezeték áthidalás             |
|            | felsovezetek | Felsővezeték átfeszítés            |
| megjegyzes | „szöveg”     | Megjegyzés (opcionális)            |

# TÉRINFORMATIKAI ALKALMAZÁSFEJLESZTÉS

Az összegyűjtött és feldolgozott adatok megjelenítéséhez szükséges webalkalmazást JavaScriptben írtam. Ennek előnye, hogy HTML fájlba ágyazható, így a megjelenítéshez csak a webcímet kell beírni a böngésző címsorába, ezután letöltés nélkül fut a programkód.

Az elkészítés során a Leaflet nevű nyílt forráskódú, JavaScript függvény-gyűjteményt használtam (Agafonkin, 2010.), ami ingyenesen letölthető és szabadon használható. Segítségével téradatok megjelenítésére alkalmas, weboldalba ágyazható térképek hozhatók létre. A könyvtár elérési útját csupán a HTML fájl **head** részébe kell beilleszteni a megfelelő **script** címkék közé. (Gede, Script nyelvek alkalmazása a web-kartográfiában, 2020)



A webalkalmazás terjedelme miatt (több, mint 1200 sor) az alkalmazás elemeinek bizonyos funkciói nem mindig kerülnek teljes egészben bemutatásra.

## A térképobjektum

A HTML fájl törzsében egy egyedi azonosítóval ellátott **div** elemet hoztam létre, aminek a stíluslapját úgy állítottam be, hogy kitöltse a teljes rendelkezésre álló helyet:

```
<div id = "terkep_helye"></div>
```

A **script** részben ehhez az elemhez rendeltem hozzá a térkép helyéül szolgáló interaktív területet, egyben megadva a maximálisan engedélyezett nagyítás mértékét, és a betöltéskor alapértelmezetten megjelenítendő hely földrajzi koordinátáit, nagyításértékét:

```
var map = L.map('terkep_helye',{maxZoom: 20})  
.setView([47.43421, 19.08799], 17);
```

## Háttértérképek

A térképobjektum létrehozása önmagában csupán egy geoinformatikai adatok befogadására, kezelésére és megjelenítésére szolgáló üres dobozt eredményez az oldalt egy böngészőben megnyitva. Háttértérképként egy várostérkép és egy műholdkép választható a rétegkezelőben.

A várostérkép létrehozásához a MapTiler szolgáltatást vettem igénybe (Pridal, 2008.), aminek segítségével jellemzően webre szánt háttértérképeket lehet készíteni különböző



tematikákkal. A megjelenés széles körűen testreszabható, a szükséges rétegek kiválasztásától kezdve egészen a rajzolatukig bármit be lehet állítani. Az elkészített térképet a rendszer a felhasználó saját fiókjába menti el vector tile-réteggént<sup>3</sup>, amihez hivatkozást is generál. A MapTiler JavaScript-függvénykönyvtárának letöltése és az arra való hivatkozás **head** részbe illesztése után a scriptben egy változóban adjuk meg az elkészített térkép elérési- és formázási hivatkozását.

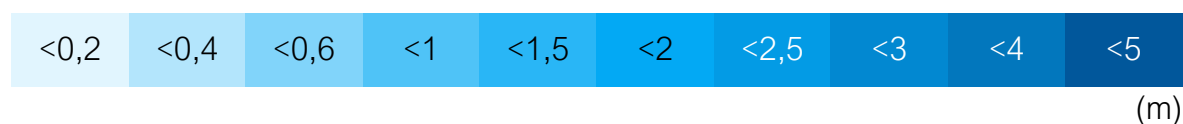
A műholdkép esetében a Google felvételét használtam (Maxar Technologies, 2021.), a HTML fájlhoz hozzáadás az előzőekhez hasonlóan zajlott.

## Medermélység WMS-réteg

A medermélységeket tartalmazó „*depth\_3857.shp*” fájlra hivatkozva a MapServer eszközeinek segítségével (University of Minnesota, 1994.) egy mapfile-t készítettem, mivel az állományt méretéből adódóan nem lehetne hatékonyan kezelni vektoros formában.

A MapServer egy Web Map Service (továbbiakban WMS) típusú szerveroldali alkalmazás, ami téradatok megjelenítését teszi lehetővé webes alkalmazásokban. Sokféle fájlformátumot képes értelmezni, többek között az ESRI által kifejlesztett shapefile állományokat is kezeli. A program az utasításokat egy — a szerverre feltöltött — mapfile segítségével hajtja végre. A mapfile egy szöveges leíró adatokat tartalmazó dokumentum, amiben meg kell adni a forrásfájl elérési útját, a téradatok vetületét, a kivágat koordinátáit, a megjelenő rétegeket és a kimeneti fájl formátumát és típusát. Ezeken kívül sok réteg- és formázási beállításra van lehetőség. (Gede, A MapServer használata, 2020)

Tíz medermélység-osztályt hoztam létre a mapfile-ban az „*elevation*” attribútum alapján, ezekben mindig a legszűkebb feltétel érvényesül. Az osztályok bemutatása után a mapfile medermélység-rétegére vonatkozó részlete helyezkedik el a metaadatok nélkül.



---

<sup>3</sup> Vector tile-rétegek esetében a betöltött térképet általában egy külső szerver szolgáltatja. Ezeket az adatokat vektoros felépítésű állományokban tárolja a szerver, a kliens gép kérésére csupán az aktuálisan megjelenítésre kerülő területhez szükséges „csempékre” darabolt részeket küldi a hálózaton keresztül. A vektoros adatok általában jóval kisebb tárhelyigényűek raszteres társaiknál, így az adatátvitel is gyorsabb. A kapott adatok képpé alakítását már a kliens gép végzi, így a megjelenítés gyorsasága nagyban függ a végfelhasználó eszközének teljesítményétől.

```

LAYER
  NAME "depth_3857"
  TYPE polygon
  DATA "depth_3857.shp"
  STATUS on
  PROJECTION
    "init=epsg:3857"
  END
  (...) Metaadatok

  CLASS
    NAME "0.2"
    EXPRESSION ([ELEVATION]<=0.2)
    STYLE
      COLOR 225 245 254
    END
  END
  (...) A többi osztály
END

```

A mapfile-t feltöltöttem az ELTE Mercator szerverére, ami futtatja a MapServer alkalmazást, így sikerült képfájlá alakítani a medermélység-adatokat.

A Leaflet függvénykönyvtárban van lehetőség WMS alkalmazások által generált képfájlok raster tile-réteggént való megjelenítésére is, sőt ezeken a rétegeken lekérdezéseket is el lehet végezni az [L.WmsGetFeatureInfo](#) metódus segítségével. Így került fel a medermélység is térképréteggént a webalkalmazásba.

## *GeoJSON rétegek*

A hatalmas tárhelyigényű fájlok betöltése gyorsabb WMS-réteggént, azonban ügyelni kell, hogy ne legyen feleslegesen sok belőlük, mert ilyenkor a térkép betöltése lelassulhat. A többi réteget, mivel nem álltak akkora adathalmazból, mint a medermélység, a könnyebb kezelhetőség érdekében GeoJSON formátumba konvertáltam a QGIS-ből.

A GeoJSON állományok alapja a JSON (JavaScript Object Notation) formátum. Mint a nevéből kitűnik, ez egy JavaScripthez kifejlesztett objektumleíró fájltypus, amivel egyszerű adatstruktúrák írhatóak le, és amit azóta számos más területen is használnak. Előnye, hogy a fájl forráskódjának értelmezéséhez nincs szükség számítógépre, az adatok attribútumpárokban, illetve az attribútumhoz kapcsolt tömbökben foglalnak helyet rendezett szöveges formában.

A GeoJSON fájlban az egyes adathalmazokhoz térbeli leíró attribútumadatok is tartoznak. Rendezett koordináta-párok pontokat adnak meg a földfelszínen, ezeket tömbökbe ágyazva vonalláncok vagy poligonok hozhatók létre, valamint további tömb-szintek bevezetésével csoportosíthatóak is.

Ezeket a fájlokat a Leaflet natívan kezeli, elég csak beilleszteni a fájl tartalmát a programkód megfelelő részébe. Ha azonban nem csak néhány pontról van szó, és érdemesebb külön fájlban tárolni emiatt, természetesen van lehetőség elérési útvonallal hivatkozni rá.

Alább a Kikötőhelyek GeoJSON-réteg kódrészlete látható. Az első sorban a megadott címről betölt a kívánt fájlt, a következő sorban ezt egy változóban rögzítettem, a többi sor pedig a stílusleírásra szolgál. A kategorizált megjelenítés beállításához meg kell adni melyik attribútum alapján történjen az osztályozás (jelen esetben ez a „*tipus*” oszlop), ezt követően minden osztályhoz megadtam a beállításokat, ügyelve a megfelelő szintaxisra.

```
fetch('layers/rsdSteg.geojson').then(r=>r.json()).then(function(data) {  
var kikotohely = L.geoJSON(data, {  
  style: function(feature) {  
    switch (feature.properties.tipus) {  
      case 'steg': return {color: "#b5a785", fillOpacity: 1,  
        opacity: 0.5,};  
      // A többi kategóriához tartozó stílus  
    }  
  }  
}).addTo(map);  
retegkezelő.addOverlay(kikotohely, 'Kikötőhelyek')  
});
```

A stílusbeállítások után nem maradhat ki a Leaflet `.addTo()` metódusa; így kerül fel az adott réteg a térképre. Az utolsó sorban fedvényréteggént hozzáadjuk a rétegkezelőhöz is Kikötőhelyek címkézéssel ellátva.

Az `rsdPoi.geojson` pontrétegen található helyeknek az attribútumtáblájában további hasznos adatok találhatók, ezeket célszerű kattintásra felugró ablakban megjeleníteni. A Leaflet erre is kínál lehetőséget, a réteghez hozzáfűzhető `.bindPopup()` metódussal:

```
.bindPopup(function (layer)  
{return String(  
  "<b>" + layer.feature.properties.nev + "</b>" + "<br>" +  
  // A többi attribútumadathoz tartozó utasítások  
}),})
```

## Rétegkezelő

A Leaflet rétegkezelőhöz az alábbi kódrészletre volt szükség:

```
var alap = {'Térkép': gl, 'Műhold': googleSat};  
var retegkezelolo = L.control.layers(alap).addTo(map);
```

Az első sorban egy objektumot hoztam létre alap néven, amibe a korábban „gl” néven definiált MapTiler térképréteget és a „googleSat” néven definiált Google Műholdkép réteget tettem bele. Minkettő kapott egy beszédesebb nevet; a felhasználói felületen ezek fognak majd megjelenni.

A második sorban magát a rétegkezelőt hoztam létre, és megadtam az előbb elkészített objektumot, hogy az ebben szereplő rétegek jelenjenek meg a benne. Ezek után a rétegkezelőt hozzáadtam a térképhez.

Mindez azt jelentette, hogy az említett rétegek alapréteggként kerültek fel a térképre, azaz közülük egyszerre egy kerülhet megjelenítésre, amit rádiógombokkal lehet kiválasztani.

Az úgynevezett fedvényrétegek közül tetszőleges számú lehet egyszerre aktív, megjelenítésüket jelölőnégyzetekkel lehet szabályozni. Ezeknek a rétegeknek a vezérlői a későbbiekben kerültek bele a rétegkezelőbe. A hozzáfűzéshez minden fedvényréteg definiálása után egy sor kellett, amiben rögzítjük, hogy fedvényréteget adunk a rétegkezelőhöz, megadjuk a réteg nevét és a szöveget, ami a felhasználó számára látható lesz. Az alábbi példában a „kikotohely” néven definiált Kikötőhelyek réteg szerepel.

```
rettegkezelolo.addOverlay(kikotohely, 'Kikötőhelyek')
```

## Helymeghatározás

A terepi tájékozódás elősegítése miatt készítettem az alábbi programrészletet, aminek segítségével a térképen megjelenítésre kerül a programot futtató eszköz földrajzi helyzete. Ebben segítségemre volt a Stack Exchange programozási fórumának egy hasonló témájú kérdése. ("Poly Geo", 2016.)

Első lépésben „helyIkon” néven definiáltam egy Leaflet ikon objektumot, aminek megadtam az elérési útját (maga a képfájl is saját készítésű), és a méreteit.

```
var helyIkon = L.icon({
```

```

        iconUrl: 'http://mercator.elte.hu/~jki9y6/RSD/pics/location.png',
        iconSize: [30, 30]
    });

```

Ezt követően definiáltam egy-egy globális változót, a helyjelzőnek és a helymeghatározás hibahatárát jelző körnek.

```

var locmarker = null; var loccircle;

```

Az `onLocationFound()` függvény akkor fut le, ha az eszköz megtalálta a helyet. Ekkor megjelenít egy gombot a kezelőfelületen visszajelzésképp, amennyiben nem volt előzőleg meghatározott hely, felkerül a térképre a helyjelző és a hibahatár jelző kör. Ha volt előzőleg meghatározott hely, úgy módosításra kerülnek ezek az objektumok.

```

function onLocationFound(e) {
    document.getElementById('helyesgomb').style.display="block";

    var radius = e.accuracy / 2;
    var loc = e.latlng

    if (locmarker == null) {
        locmarker=L.marker(loc, {icon: helyIkon}).on('click',
            delLocation).addTo(map);
        loccircle=L.circle(loc, radius).addTo(map);
    }
    else{
        locmarker.setLatLng(loc);
        loccircle.setLatLng(loc);
        loccircle.setRadius(radius);
    }
}

```

Ha az eszköz valamilyen okból kifolyólag nem talál helyadatokat, az `onLocationError()` függvény fut le, és egy hibaüzenetet ír ki a képernyőre.

```

function onLocationError(e) {
    alert(e.message);
}

```

Azt, hogy az előbbi két függvény közül melyik lépjen életbe, a `getLocationLeaflet()` függvény dönti el, az alapján, hogy sikeres volt-e a helymeghatározás.

```

function getLocationLeaflet() {
    map.on('locationfound', onLocationFound);
    map.on('locationerror', onLocationError);
    map.locate({setView: true, maxZoom: 16});
}

```

```
closemenu();
}
```

A `delLocation()` függvény a helyjelzőre kattintva fut le, és letörli azt és a hozzá tartozó hibahatár jelző kört, valamint mindkét változó értékét `null`-ra állítja vissza. Ezen felül a helymeghatározás sikerességét jelző gombot is eltünteti a kezelőfelületről.

```
function delLocation() {
    map.removeLayer(locmarker);
    map.removeLayer(loccircle);
    locmarker = null;
    loccircle = null;
    document.getElementById('helyesgomb').style.display="none";
}
```

A helymeghatározás folyamatát a `getLocationLeaflet()` függvény fogja elindítani, ami a helygomb azonosítójú gombra kattintva fut le.

```
<img src = "imagemap/gps.png" title = "Aktuális helyzet"
onClick = "javascript:getLocationLeaflet();" id = "helygomb">
```

## Kezdőnézet

A térkép betöltésekor a kezdőnézet a Gubacsi híd környékén van. Itt található a Kolonics György Vízisport Központ, ami Soroksári-Duna legnagyobb és legjelentősebb vízisport-telepe, egyben a Magyar Kajak-Kenu Szövetség budapesti edzőközpontja is.

Annak érdekében, hogy erre a központi helyre gyorsan vissza lehessen navigálni, a `div` elemekből létrehozott menüben egy  Térképnyíl alaphelyzetbe állítása gomb típusú `input` elemet hoztam létre, amely kattintásra visszaállítja a kezdőnézetet és a nagyítást, ezzel együtt bezárja a menüt.

```
function locateHome() {
    map.setView([47.434214, 19.087993], 17);
    closemenu();
}
```

## Áttekintőtérkép

A Soroksári-Dunán való gyors böngészés elősegítésére egy áttekintőtérképet készítettem, amin a part menti települések hivatkozások szerepelnek. Ezekre kattintva a főtérkép középpontja a kiválasztott helyre ugrik, és egy fix nagyítási szint kerül beállításra.

Az áttekintőtérképet Microsoft Publisherben készítettem zömmel téglalapok és szabadkézi alakzatok felhasználásával, majd képként mentettem el. A képet egy imagemap<sup>4</sup> készítő weboldalra töltöttem fel, bejelöltem a gombok helyét. Az ezek alapján relatív módon generált hivatkozásokat tartalmazó HTML kódrészletet elmentettem.



3. ábra: Az áttekintőtérkép

A `body` címkén belül létrehoztam egy `div` elemet, amibe beágyaztam a képet és beillesztettem az elmentett imagemap-kódrészletet. A bemutatott kódrészletben nem szerepel egyenként az összes városhoz tartozó hivatkozás, mert ugyanazon séma alapján épülnek fel:

<sup>4</sup> A HTML lehetővé teszi, hogy képekhez hozzárendeljünk téglalap és kör alakú hivatkozási területeket, így a kép bizonyos részei kattinthatóvá válnak. Az imagemap tartalma megadható abszolút módon koordinátákkal, illetve relatív módon a kép oldalaihoz viszonyított aránnyal, százalékban kifejezve. Relatív megadással a kép átméretezése esetén sem „romlik el” az elkészített imagemap.

```

<div id = "attekintoterkep">
  
    (...) A többi hivatkozási terület megadása
    <a onclick="javascript:kvassayzsilip();" style="width:25.3%;
    height:5.5%; left:30.7%; top:1.5%; position:absolute; cursor:
    pointer; display:block; z-index:5; overflow:hidden;"></a>
    <input type = "button" title = "Bezárás" value = "🌐 Bezárás"
    onClick = "javascript:hideOverview();" id = "attekintobezaras">
</div>

```

Hozzáadtam továbbá egy gombot amire kattintva bezárható maga az áttekintőtérkép.

A hivatkozások működéséért JavaScript függvények felelősek. Azon túl, hogy a megfelelő helyre állítják a térképet, bezárják az áttekintőtérképet, így az a továbbiakban nincs útban. Példaként a Kvassay-zsiliphez tartozó függvény, a többi itt is ugyanilyen módon van megírva:

```

function kvassayzsilip() {
  map.setView([47.461359, 19.071478], 15);
  document.getElementById('attekintoterkep').style.display="none";
}

```

Az „attekintobezaras” azonosítójú `input` elem (a bezárás gomb) működéséért felelős `function hideOverview()` függvény az előbb bemutatott hivatkozásokhoz tartozó függvénynek csupán a második sorát tartalmazza.

## Vízi útvonalterv

A webalkalmazás legösszetettebb része és az egyik fő lényegi eleme a két választott pont között tervezhető vízi útvonal, ami több beállítással is bír. Elkészítésében témavezetőmtől sok gyakorlati segítséget kaptam.

Az útvonaltervezést végrehajtó programkód-részlet 145 sorból áll, így csak a CD-mellékletben található meg. Itt a funkciók és a lényeg ismertetése következik:

Az  *Útvonaltervezés* gombra kattintva indul el a folyamat és amíg aktív, a gomb átvált az  *Útvonaltervezés megszakítására*. Egy megjelenő `div` elemen olvasható, hogy a kezdőpont a térképre kattintással adható meg, ami után egy újabb `div` elem értelemeszerűen a végpont ugyanilyen módon történő megadásáról informálja a felhasználót. A pontok megadása csak a Dunaágon, és annak meghatározott környezetében lehetséges. Ha ennél messzebb történik a kattintás, egy felugró ablak figyelmeztet a hibára.



Az útvonal kezdő- és végpontjának megadását követően az útvonaltervezés egy rejtett GeoJSON linestring objektum — a Soroksári-Duna középvonala — mentén automatikusan lefut, és kirajzolódik a térképen.

A folyamat közben a háttérben létrejön egy tömb, aminek a nulladik tagja a kezdőpont középvonalra vetített pontjának koordinátáit kapja meg értéként, majd a program a középvonal réteg szélességi koordinátáját addig növeli egy ciklusban, amíg még nem éri el a végpont szélességi koordinátáját. Ez azért lehetséges ilyen módon, mert a Soroksári-Dunán sodrással lefelé haladva helyzetünk szélességi koordinátája szigorúan monoton csökken. Az így megkapott középvonalra illesztett szakasz hossza adja meg az útvonal távolságát.

A felhasználói felületen megjelenik egy `div` doboz útvonaladatokkal és beállításokkal, ami még sok HTML-elemet tartalmaz. Ezen olvasható az útvonal hossza, és becsült időtartama. A becsült időtartamot a program hajótípustól, az evezés tempójától, és attól függően számítja, hogy sodrással vagy azzal szemben történt a tervezés.

Kétszemélyes túrakenu

Evezési tempó

Felfedező evezés Kényelmes, folyamatos tempó Erőltetett menet

Az útvonal teljes hossza: 15.97 km  
Időtartam: 126 perc

OK

- ☒ Kétszemélyes túrakenu
- ☐ Négyszemélyes túrakenu
- ☐ Túrakajak egyes
- ☐ Túrakajak páros
- ☐ Sárkányhajó
- ☐ SUP deszka

4. ábra: Útvonaladatok ablak (balra) és a hajótípusválasztó felugró lista (jobbra)

A hajótípus egy felugró listából választható ki, az evezés tempója pedig egy háromállású csúszka input elemen állítható. A sarokban található „OK” gombbal bezárható az útvonaltervezés.

## Jelmagyarázat

Akár nyomtatott, akár webes térképről van szó, a jelmagyarázat elengedhetetlen elem. Elkészítéséhez — akárcsak az áttekintőtérkép esetében — a Microsoft Publishert használtam. Minden összetevő, így az egyes pontszerű elemeket jelölő képfájlok is saját készítésűek, ezeket szintén a kiadványszerkesztőben csináltam.



5. ábra: Saját készítésű markerek közül néhány, balról sorban: Csónakház, Hotel, Strand, Emlékmű

A kész Jelmagyarázatot képként mentettem, majd beágyaztam a HTML fájlba. A menüben található Jelmagyarázat gombra kattintva fut le az `openLegend()` függvény, ami bekapcsolja a Jelmagyarázat `img` elem láthatóságát, a térképobjektumra pedig egy `backdrop-filter: blur;` CSS-tulajdonságot alkalmaz.



6. ábra: Jelmagyarázat

## Formázás

A weboldalt felépítő HTML-elemek formázása nem az utolsó lépése volt a munkának. Hogy átlátható és működőképes legyen a webalkalmazás már a programkód írása alatt, az egyes elemek formázását azoknak létrehozásakor vagy a kapcsolódó függvények megírásakor el kellett kezdeni.

A HTML-elemek megjelenésének leírására a CSS<sup>5</sup>-szabvány jól használható. Szintaxisa egyszerű, ésszerű felépítés esetén könnyen átlátható és gyorsan módosítható. Az elemek egyenként és csoportosan is formázhatók, valamint megjelenési tulajdonságaik bizonyos eseményekre (pl.: kurzor elem fölé helyezése, mobil eszközön való megjelenés, függvény hatása) dinamikusan változhatnak.

A CSS formázást a HTML fájl `head` részén belül kell megadni `style` címkék közé illesztve. Másik lehetőség külön szöveges fájlba menteni .css kiterjesztéssel, ilyenkor a `link` címkék közé kell a hivatkozást beilleszteni és a `rel="stylesheet"` tulajdonság megadásával jelölni, hogy stíluslapról van szó.

Az elemek megfelelő formázási beállításai az egész weboldal forráskódjának megközelítőleg a felét teszi ki a sorok számának tekintetében.

## A felhasználói felület egyéb elemei

A HTML-elemek CSS formázásán túl az egységes megjelenés érdekében nem használtam külső forrásból letöltött képeket, ábrákat és ikonokat. A gombokon és kezelőszerveken látható piktogramokat egyszerű alakzatokból készítettem, és .png formátumba mentettem, mivel az támogatja az egyes képpontokhoz rendelhető áttetszőség értékeket, így háttér nélkül használhatóak lesznek.



7. ábra: A kezelőszerveken megtalálható egyszerű ikonok biztosítják a rugalmas megjelenítést

---

<sup>5</sup> Cascading Style Sheets (egymásba ágyazott stíluslapok): Legtöbbször HTML típusú strukturált dokumentumok megjelenését írja le. Segítségével meghatározható, hogy hogyan jelenjenek meg az egyes HTML elemek. Többek között befolyásolható a színük, méretük, elhelyezkedésük, margóik stb. Az egymásba ágyazhatóság arra utal, hogy több stíluslap, meghatározás is megadható egyszerre, illetve egy stílus lehet több elemre is érvényes, ami egy másik stílussal felüldefiniálható.

## EREDMÉNYEK, TOVÁBBI CÉLOK

Jelenlegi állapotában kialakítottam az egész Soroksári-Duna interaktív webtérképéhez használható alapot: fájlrendszert, attribútumtábla-felépítést, és azt a webes funkcionalitást, ami lehetővé teszi, hogy a mintaterületen túlhaladva már további adatigénylés nélkül, csupán vízi és part menti helyszíneléssel a térkép feltölthető legyen adatokkal. A mintaterület 11 kilométeres szakaszán pedig lehet modellezni a használatot, eszközölni az esetlegesen további felmerülő igényeket, funkcionalitást érintő fejlesztéseket.

A Soroksári-Duna webalkalmazás az alábbi hivatkozáson érhető el:

 <https://bakotamas.github.io/rsd/>



Jelen dolgozat leadásakor (2021. május 15.) a HTML fájl és a programkód aktuális állapotát rögzítő fájlt CD-mellékletben csatoltam. A weben elérhető oldal tartalma ettől eltérhet, mivel a projekt túlmutat a Diplomamunkám szabta kereteken.

Távlati cél a felmérések kiterjesztése a Soroksári-Duna teljes hosszára, a webalkalmazás továbbfejlesztése, és egy arra épülő mobilalkalmazás elkészítése.

A projekt előkészítése már megtörtént, a teljes folyóág a mintaterületen kívül még hat másik, hasonló hosszúságú felmérési szakaszra van felosztva, amit akár a későbbiekben földtudományi/térképész hallgatók terepgyakorlat tárgy keretében is végezhetnek.

## ÖSSZEGZÉS, KONKLÚZIÓ

A munkafolyamat során számos térinformatikai, és webes eszközt használtam. A téradatok feldolgozásához a QGIS volt segítségemre, a digitalizálást és a kapott adatok feldolgozását is itt végeztem, az adatokkal ESRI shapefile formátumban volt célszerű dolgozni, majd később konvertálni ezeket GeoJSON fájlá. A medermélység ábrázolásához a MapSertvert használtam, a webtérkép írása közben egyre több mindent sikerült elsajátítanom a JavaScript programnyelv nyújtotta lehetőségekből, és nem utolsó sorban a CSS-formázás kapcsán is számos újdonsággal találkoztam.

A webfejlesztés és webes kartográfia terén szerzett új ismeretek minden bizonnyal hasznomra válnak későbbi tevékenységeim során.

# ÁBRA- ÉS TÁBLÁZATJEGYZÉK

## Ábrajegyzék

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| 1. ábra: A Tematikus tartalmat érintő információcsökkenési tendencia, balról jobbra: 1930, 1970, 1980 ..... | 7  |
| 2. ábra: Topológiai hibák kijavítása .....                                                                  | 10 |
| 3. ábra: Az áttekintőtérkép.....                                                                            | 23 |
| 4. ábra: Útvonaladatok ablak (balra) és a hajótípusválasztó felugró lista (jobbra) .....                    | 25 |
| 5. ábra: Saját készítésű markerek közül néhány, balról sorban: Csónakház, Hotel, Strand, Emlékmű.....       | 26 |
| 6. ábra: Jelmagyarázat .....                                                                                | 26 |
| 7. ábra: A kezelőszerveken megtalálható egyszerű ikonok biztosítják a rugalmas megjelenítést.....           | 27 |

*Az első ábra forrása: (Faragó, 2020). A dolgozat minden további ábrája saját készítésű.*

## Táblázatjegyzék

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| 1. táblázat: Az RSD_kikotohelyek.shp fájl attribútumtábla-felépítése ..... | 11 |
| 2. táblázat: Az RSD_partszakaszok.shp fájl attribútumtábla-felépítése..... | 12 |
| 3. táblázat: Az RSD_novenyzet.shp fájl attribútumtábla-felépítése.....     | 12 |
| 4. táblázat: Az RSD_poi.shp fájl attribútumtábla-felépítése .....          | 13 |
| 5. táblázat: Az RSD_alappont.shp fájl attribútumtábla-felépítése .....     | 14 |
| 6. táblázat: Az RSD_hidak.shp fájl attribútumtábla-szerkezete .....        | 15 |

*A dolgozat minden táblázata saját készítésű.*

## IRODALOMJEGYZÉK

- "Poly Geo". (2016.. 10 20). *Stack Exchange*. Letöltés dátuma: 2021.. 05 11, forrás:  
<https://gis.stackexchange.com/questions/182068/getting-current-user-location-automatically-every-x-seconds-to-put-on-leaflet>
- Agafonkin, V. (2010.). *Leaflet*. Letöltés dátuma: 2021. 5. 3., forrás: <https://leafletjs.com/>
- Faragó, I. (2020). *Térképkiadványok - Vízisport-térképek*. Faragó Imre, Budapest.
- Gede, M. D. (2020). *A MapServer használata*. Letöltés dátuma: 2021. 05 12, forrás:  
<http://mercator.elte.hu/~saman/hu/okt/mapserver/>
- Gede, M. D. (2020). *Script nyelvek alkalmazása a web-kartográfiában*. Letöltés dátuma:  
2021. 05 12, forrás: [http://mercator.elte.hu/~saman/hu/okt/script\\_jegyzet.html](http://mercator.elte.hu/~saman/hu/okt/script_jegyzet.html)
- Kartográfiai Vállalat. (~1970, ~1980). *A Duna vízisport-térképe Esztergamtól Dunáújvárosig*.
- Magyar Királyi Állami Térképészet. (1930). *Vízi Sporttérképek - Duna Vác-Eszterom*.
- Maxar Technologies. (2021.). *Maxar Technologies*. Letöltés dátuma: 2021.. 05 11, forrás:  
<https://www.maxar.com/>
- Pridal, P. (2008.). *MapTiler*. Letöltés dátuma: 2021. 5. 3., forrás: <https://www.maptiler.com/>
- University of Minnesota. (1994.). *MapServer*. Letöltés dátuma: 2021. 5. 3., forrás:  
<https://mapserver.org/>

## KÖSZÖNETNYILVÁNÍTÁS

Köszönettel tartozom tanáromnak és témavezetőmnek, Dr. Gede Mátyásnak a diplomamunkám témajavaslatát illetően, valamint a konzultációk alkalmával nyújtott sok gyakorlati útmutatásért.

Továbbá az Országos Vízügyi Főigazgatóság Műszaki Téradat Szolgáltató Főosztály dolgozóinak, a részemre engedélyezett és elküldött medermélység-adatokért, és edzőtársamnak, Zsidai Máténak a vízi felmérésben nyújtott segítségéért.

Hálás vagyok a családomnak, valamint edzőimnek a KSI-ben, hogy végig tolerálták egyetemi tanulmányaim nappali rendszerben történő elvégzését.



## MELLÉKLETEK

Engedély



### MŰSZAKI TÉRADAT SZOLGÁLTATÓ FŐOSZTÁLY

3/a. sz. melléklet

#### ADATSZOLGÁLTATÁSI KÉRELEM TELJESÍTÉSE

Cím: **Bakó Tamás**  
Tisztelt Adatigénylő!

Hivatkozással a 2020. november 23. napján érkezett **közérdekű adat megismerésére irányuló igényének** az információs önrendelkezési jogról és az információszabadságról szóló **2011. évi CXII. törvény 29. § (1) bekezdése alapján az alábbiak szerint teszek eleget:**

Az adatszolgáltatás tartalma: **Medermodell raszteres állományban a Ráckevei-Soroksári Duna-ágról**

Az adatszolgáltatásért fizetendő költségtérítése: **költségmentes**  
Kiegyenlítendő az adatszolgáltatási kérelem visszaigazolásában felsorolt feltételek elfogadását követő 15 napon belül vagy a számla Adatigénylőhöz érkezését követő 15 napon belül.  
A költségtérítés megállapítása a 2011. évi CXII. törvény és a 301/2016. (IX. 30.) Korm. rendelet rendelkezéseinek figyelembevételével történt.

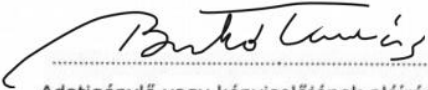
Kelt: Budapest, 2020. november 25.

  
osztályvezető  
Adattári Osztály

#### ÁTVÉTELI ÉS TELJESÍTÉSI NYILATKOZAT

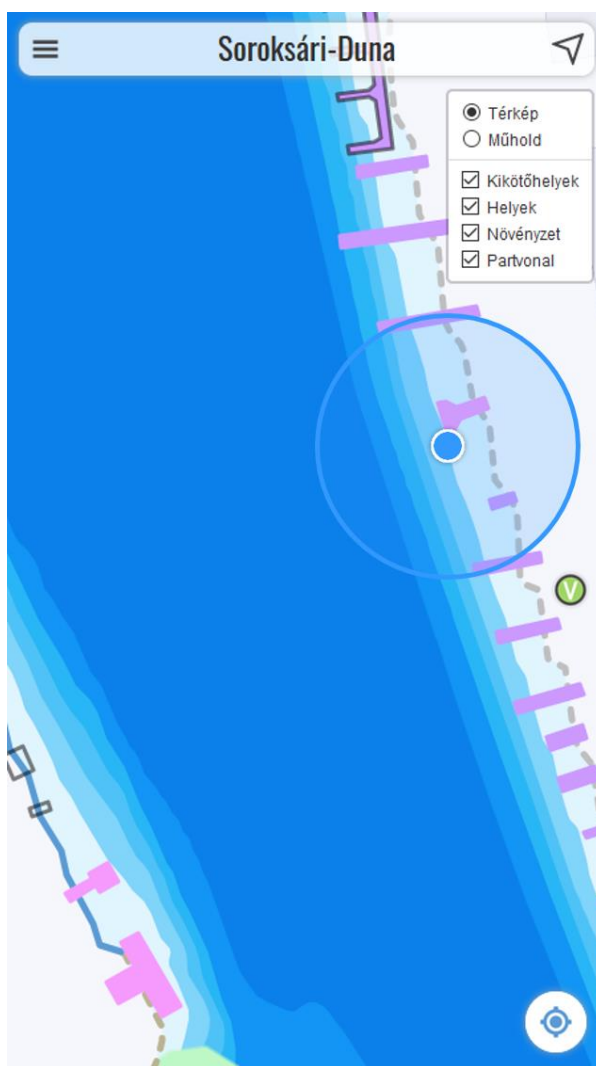
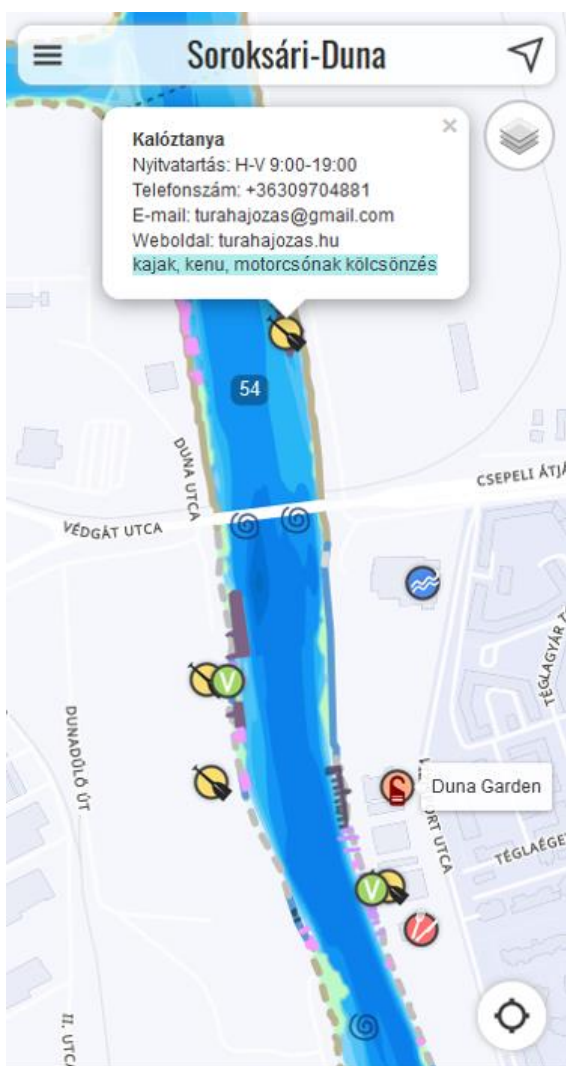
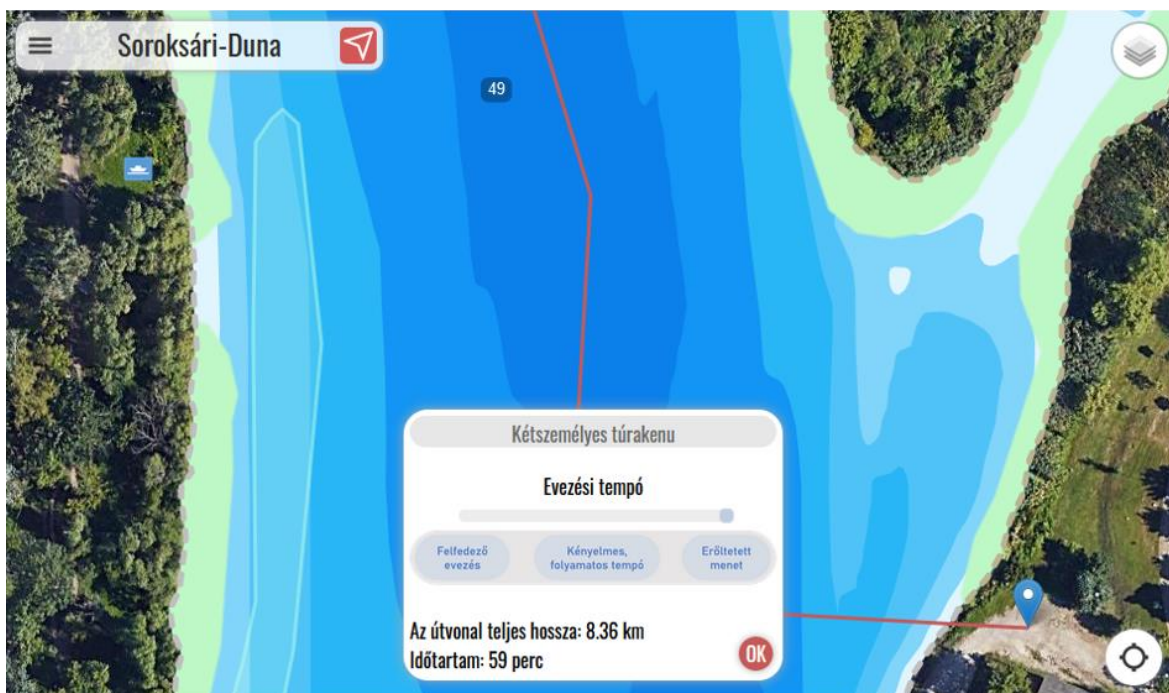
Alulírott, **Bakó Tamás**, mint Adatigénylő, illetve az Adatigénylő képviselője nyilatkozom, hogy az Országos Vízügyi Főigazgatóságtól (1012 Budapest, Márvány utca 1/d.) a mai napon a **20025-0448/2020.** számú adatigénylésünkben foglalt **„Medermodell raszteres állományban a Ráckevei-Soroksári Duna-ágról,”** tartalmú adatszolgáltatást átvettem. Átvétel előtt az adatszolgáltatás tartalmát megismertem, és igazolom, hogy az Adatkezelő által visszaigazolt adatigénylés teljesítésre került / nem került teljesítésre.

Kelt: Budapest, 2020. év 11. hó 27. nap

  
Adatigénylő vagy képviselőjének aláírása

A MI VÍZÜGYÜNK

## Képernyőképek az asztali- és mobil webes felületről



## *A CD-melléklet fájlrendszere*

|                                                                                   |                                                                                   |                                                        |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------|
|  | Adatgyűjtés .....                                                                 | A vízügyi adatok és az igénylési dokumentumok          |
|  | Térkép.....                                                                       | A shapefile munkarétegek, és a QGIS projektfájl        |
|  | WEB.....                                                                          | A webalkalmazást tartalmazó mappa                      |
|                                                                                   |  | imagemap ..... Képek, ábrák, ikonok és áttekintőtérkép |
|                                                                                   |  | layers ..... GeoJSON rétegfájlok                       |
|                                                                                   |  | leaflet ..... Leaflet függvénykönyvtár                 |
|                                                                                   |  | index.html ..... A weboldal                            |
|  | Diplomamunka.pdf.....                                                             | Diplomamunka                                           |

# DIPLOMAMUNKA EREDETISÉG NYILATKOZAT

Alulírott **Bakó Tamás**

Neptun-kód: JK19Y6

ezennel kijelentem és aláírással megerősítem, hogy az Eötvös Loránd Tudományegyetem Informatikai Karának, Térképtudományi és Geoinformatikai Intézetében írt,

## **A Soroksári-Duna interaktív vízisport-térképe**

című diplomamunkám saját, önálló szellemi termékem; az abban hivatkozott szakirodalom felhasználása a szerzői jogok általános szabályainak megfelelően történt.

Tudomásul veszem, hogy diplomamunka esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

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

HOZZÁJÁRULOK

NEM JÁRULOK HOZZÁ

Budapest, 2021. május 15.



*hallgató aláírása*