**2019.07.01. Hétfő:**

Első projektmegbeszélés. Érdeklődtem a Java rész és a Front-end iránt. Tisztáztuk a jelenléti és egyéb követelményeket, és a használandó Technológiákat. (SAP HANA)

**2019.07.02. Kedd:**

Terv: Tutorial-okat nézni, megismerkedni az Online Platformmal. Java alkalmazás készítésének utánanézése, és a program összekötése az SAP-val. Ha esetleg marad idő, Angular-nak utánanézni, kipróbálni.

Haladás: Számos kezdő, és középhaladó oktatóanyagot átnéztem és kipróbáltam köztük: Java, Java Web, Angular, Node.js. Megnéztük közösen hogy lehet adatbázist létrehozni. Az egyik tutorialt követve megkíséreltem beüzemelni egy Eclipse IDE-t hogy az képes legyen kommunikálni a SAP-al, de valamilyen port probléma miatt nem tudja letölteni a dependenciákat. Holnap ha ezt sikerül megoldanom megpróbálom összekapcsolni egy adatbázissal. (Esetleg Hibernate)

A nap végén Ismerkedtem az Angularral, de arra már csak kevés idő jutott.

**2019.07.03. Szerda:**

Terv: Megkérdezni, mi a projekt, git, todo kezelő? Port probléma megoldása, Online Spring kipróbálása, komplikáltabb java applikációt kipróbálni a platformon. Megtanulni JS-ezni.

Haladás: Adatbázishoz, SAP HANA-hoz kapcsolódó fogalmakról volt szó. Elkezdtünk gondolkodni azon hogy különböző platformokon hogyan valósítjuk meg a geo vizualizáció.

Felmerült hogy bitmap-et készítünk, és azt jelenítjük meg, hogy megjelenítjük a térképet és rá rajzolunk, pontokat, amikre ha kattintunk átirányít, továbbá felmerült külső 3rd party app bevonása. Haladtam tovább a tutorialokkal, elkezdtem megismerkedni a JS-el, és a SAPUI5-al. Elolvastam az újonnan kapott tutorialokat, HANA XS…

Megpróbáltam a Java Web Application-ra (Servlet működésének utánanéztem) komolyabb folyamatokat deploy-olni.

**2019.07.04. Csütörtök:**

Terv: Folytatni az új tutorialokat. Megpróbálni összehozni egy adatbázist egy egyszerű frontend lekérdezéssel, kapcsolat tesztelése.Java REST Controllerrel foglalkozó tananyag elvégzése, mivel szükség kell REST API-ra a szoftver fejlesztése során is.

Haladás: A reggel folyamán nekiláttam a tervezett tananyagnak, majd szoftveres problémába ütköztünk, ugyanis a SAP Cloud Service leállt, és nem tudtunk Buildelni adatbázist. Megkaptuk a mini projektet így mindenki elkezdett egy részével foglakozni. Volt aki a geokódolás után kutatott, volt aki front-endet kezdett írni. Mi az adatbázis feltöltésével próbálkoztunk viszont tekintettel arra hogy nem volt elérhető, megfelelő sémájú deployol-ható adatbázisunk, ezt nem tudtuk megtenni. Megoldásként az ELTE aramis adatbázis szerverén csináltunk egy táblát (ABIOWE.SAP) aminek köszönhetően páran neki kezdhettek a szükséges SQL kódok elkészítésének, hogy előállítsuk a szükséges kimutatásokhoz szükséges táblákat. Elkezdtem kutatni hogy hogyan érhető el az adatbázis kliensből, továbbá utánanéztem az ODATA-nak.

**2019.07.05. Péntek:**

Terv: Java kliens elkészítése és ResultSet-be összegyűjteni a lekért adatokat a szerverről (jdbc). Az ezzel kapcsolatos 2 tutorial végigolvasása (kb 3 óra). ODATA-ba beleásni magam (online/offline ?) Alkalmas Java kliens létrehozása, esetleg Java Web app deployolása.

Haladás: Ma végre sikerült Build-elni az adatbázist, így eléjük az üres táblát a megfelelő sémával, viszont továbbra sem tudunk adatot importálni a táblába, mert a felület nem működik jól..

Új tev: SAP HANA Egyetemi szervert használni, feltölteni a táblákat, onnan lekérni JSON-ba, és azt feldolgozni.

a) a teljes táblát JSON-a alakítani, és saját lekérdezéseket írni.

b) SAP oldalon megírni a lekérdezéseket/ létrehozni az új táblákat, és azt alakítani JSON-é. Ha ez megvan a front-end a JSON-t dolgozza fel, és minden kliens valamilyen segéd táblával feldolgozza/vizualizálja az adatokat. Minimum követelmény szint: <http://www.orszagleltar.gov.hu/faces/UI-tars2?_adf.ctrl-state=15mtndmp4o_3>

Később: Ha van működő SAP szerverünk, akkor ODATA segítségével fogjuk lekérdezni az adatokat, és a kliens tartalmazni fog egy StringBuildert szerű metódust aminek segítségével le fogunk tudni kérdezni adatot SQL szerű lekérdezéssel.

Azt a tanácsot kaptuk, hogy kezdjünk el dolgozni JSON-al aztán a későbbiekben ha működni fog az ODATA akkor álljunk át rá.

**2019.07.08. Hétfő:**

Terv: JSON adatok feldolgozása Java kliensen, tömbök feltöltése Java Reflexion segítségével.

Haladás: Összedolgozva egy másik hallgatóval, kitaláltuk, hogy Java platformon, felosztjuk a munkát. Én megoldom hogy a backend adatok sikeresen importálhatóak legyenek a szoftverbe, ő pedig megoldja hogy az adatvizualizáció jó legyen. Jelenleg egy JSONObject-eket tároló konténerből próbálom feldolgozható adat halmazzá átalakítani azt.

A Vincellér Zoltán tanár úrral átbeszéltük a tehendőket a hétre: mini projekt elkészítése (Táblák normalizálása->Odata->JSON->Geokódolás->vizualizáció) továbbá a projekt dokumentálása. A Reflexion API használata problémákba ütközött, holnap próbálkozom tovább.

**2019.07.09. Kedd:**

Terv: Odata specifikus konzultáció, Hozzáférés probléma megoldása, Elmaradt beszámoló megírása. Git beüzemelése. Java Reflexion API tanulás, debugolás.

Haladás: Átírtam sokkal hatékonyabbra a JSON beolvasást, ugyanakkor nem sikerült megoldanom hogy a magyar karaktereket is felismerje, az interneten nem találtam megfelelő megoldást rá, pedig órákon keresztül kerestem, ehhez segítséget kell kérnem valakitől. Megoldottuk hogy ODATA-ból ki tudjuk nyerni az JSON fájlt, holnap pedig megoldom hogy URL-ből tudjon olvasni, ha sikerült végre parse-olnom megfelelően.

**2019.07.10 Szerda:**

Terv: Régi JSON parser átírása működőképesre. Megoldani hogy URL-ből is tudjon olvasni, majd merge-lni a a geocode-os projekttel. Esetleg Dokumentáció írás.

Haladás: Kiderült hogy az encode-olási probléma fájl szintű így megoldható volt. Várhatóan nem fog jelentkezni amikor URL segítségével érjük el az adatbázist. Az elérés elméletileg jó ugyanakkor probléma hogy a szervernek nincsen tanúsítványa így az alkalmazás nem hajlandó elérni az ODATA-t, amiből ki tudnánk nyerni a JSON-t. Az adatfeldolgozást tökéletesítettem, és sikeresen mergel-tem Kovács Mátéval, és a lakcímek hosszas feldolgozása után, most már minden egyes címnek le tudjuk kérni a koordinátáit. Következő lépés hogy ezeket az adatokat egy API segítségével megjelenítjük egy front-end felületen. Ha ez megvan még 2-3 grafikont kéne megoldani, és egy Calculation View-et.

**2019.07.11. Csütörtök:**

Terv: 1. Java app kipróbálása Szervleten, 2. Vizualizáló API-t keresni, 3. megoldani a tanúsítvány problémát. 4. Front-end készítése

Haladás: Hosszas kínlódás után úgy döntöttem hogy a Bing ingyenes Map API-ját használom. A probléma ezzel az hogy az előre konfigurált fájlt nehéz előállítani így úgy döntöttem hogy a koordinátákat exportálom ki egy JSON fájlba amit egy JavaScript alkalmazás beolvas és feldolgoz. Erre azért volt szükség mert a Bing API-ja JS-t használ, és nem találtam módot a Java-ba való integrálásra. Ez után back-end oldalon elértem hogy a koordináták ne ismétlődjenek, pusztán optimalizálás céljából. (MIvel durván minden 3. koordináta megegyezik). Ez után a script segítségével megjelenítenem a megszerzett koordinátákat a térképen. Próbálkoztam tanúsítvány importálásával hogy URL-en keresztül el tudjam érni az OData állományt, de jelenleg nem jártam sikerrel.

**2019.07.12. Péntek:**

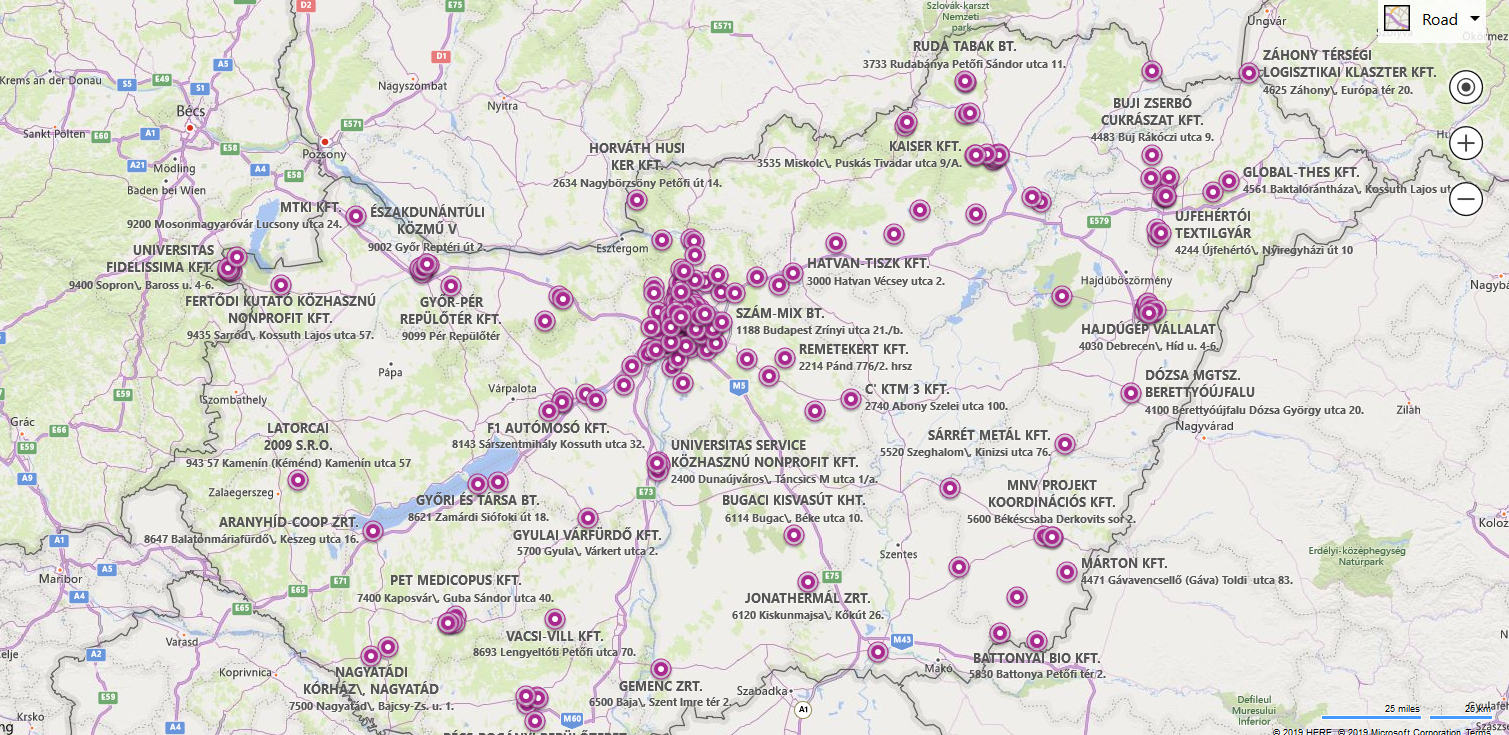
Terv: Megoldani a közvetlen ODATA elérést, esetleg hogy többet mint 1000 db-ot. Cégek nevét is kirakni Bing API-val, JFrame az app-nak, esetleg feltenni a homokozóba.

Haladás: Sikerült megoldani a vizualizálást továbbá készült egy mini JFrame felület is 2 gombbal, a funkciók működtetéséhez. (ProcessData, Visualise), továbbá sikerült elérni hogy a script a beolvasott koordináták alapján megjeleníti az összes pontot a térképen.

**A program működése tömören:**

1. Az első lépés az adatbázis elérése, ez ODATA-n keresztül történik. A Az odata-t lekéri a program JSON formátumban egy String-be majd GSON könyvtárral feldolgozza JAVA oldalon. (Jelenleg jogosultsági problémákba ütköztem, mivel a SAP HANA oldal-t nem hajlandó elérni a kliens mivel nincs tanúsítványa, és a tanúsítvány elfogadtatásához rendszergazdai jogosultsághoz lenne szükségem. Mindazonáltal a kód kész, és le van tesztelve, de jelenleg lokális .json fájlból olvassa be és dolgozza fel az adatokat, amik 100%-ban megegyeznek az URL segítségével lekérhetőkhöz.)
2. A json file parse-olása után az adatok mentésre majd feldolgozásra kerülnek, egy az adattábla sémájával megegyező struktúrájú konténerben. Innen egyszerűen és gyorsan elérhetőek.
3. A későbbiekben a Bing Map API-ját használom az adatbázisból beszerzett címek megjelenítésére. Ezzel az egyetlen probléma hogy egyszerre csak 1 címre lehet manuálisan keresni, ezért úgy döntöttem hogy még Java oldalon minden címhez lekérem a hozzájuk tartozó koordinátákat és azokat csak a lekérdezés után jelenítem meg vizuálisan a térképen. Erre szolgál egy osztály számos metódussal együtt. A getCoordinates() metódus megkapja a címet paraméterben és a Bing API egy szolgáltatása segítségével visszaadja a cím koordinátáit, majd eltárolja azokat.
4. Az idő szűkössége miatt egyszerűbbnek láttam hogy külön böngésző ablakot nyissunk a térképnek. A Font-end azonban JavaScriptet használ a megjelenítéshez, így Java oldalról a program exportálja a koordinátákat (a későbbiekben terv szerint a cég egyéb adatait is) és a böngésző megnyitásakor beolvassa azokat.
5. Ekkor már gyerekjáték megjeleníteni a koordinátákat a térképen.

Tehát a program a következő sémát követi:

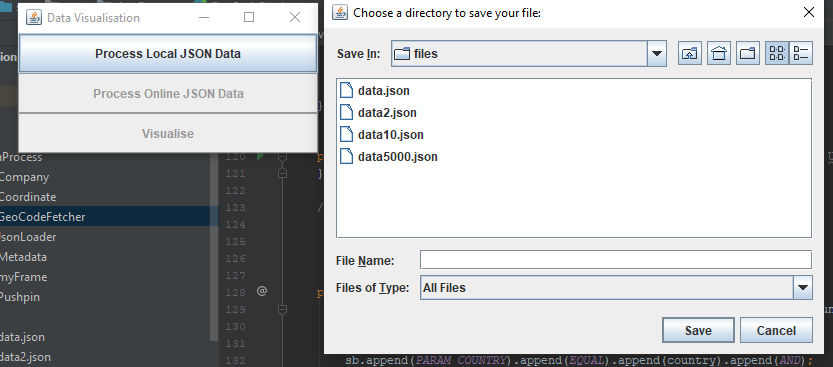
**SAP HANA → ADATBÁZIS → ODATA → JSON → JAVA → JS → BING API**

**2019.07.15. Hétfő:**

Terv: Megoldani a közvetlen ODATA elérést, esetleg hogy többet mint 1000 db-ot. Cégek nevét is kirakni Bing API-val. Indítófelület kiegészítése.

Haladás: Az ODATA állományt a szerverről, továbbra se érjük el, ez probléma, pláne hogy 1000-nél nagyobb adatothalmazt nem tudunk lekérdezni se. Otthon megpróbáltam megoldani Rendszergazdai jogosultsággal a tanúsítvány problémát, sikertelenül.

Kiegészítésre került a JFrame, immár két külön funkcióként elérhető az adatbeolvasás. Elérhető a (jelenleg, a fent említett hiba miatt) nem működő online lekérdezés, továbbá megmaradt a lokális JSON feldolgozás. Az egyszerűség kedvéért készült egy fájl kiválasztó menő, amivel tallózni lehet a feldolgozandó fájlt. Ezen felül a koordináták mellett a teljes, nyers cím és a cég rövid neve is exportálásra került a front-end felé, aminek az eredménye hogy a fenti, frissített képen látható módon megjelennek a címek, és a cégek nevei is a térképen.



**2019.07.16. Kedd:**

Terv: A Microsoft Pin-ek kattinthatóvá tétele., Dokumentáció írás majd projekt beadása GitHubra, JavaScript tanulás.

Haladás: Délelőtt, elkészítettem a projektet feltöltéshez, viszont nincs jogosultságunk a repóhoz. Fejlesztettem a térképet, és igyekeztem optimalizálni a Java kódot is. Délutánom nagy része pedig JavaScript tutorial-okkal ment el, hiszen Szerdától ha minden jól megy a Cserép Mátéval fogok dolgozni, egy JS alapú témában. Utána néztem a *Leaflet* könyvtárnak, és kezelésének. (<https://leafletjs.com/>) (<https://javascript.info>)

**2019.07.17. Szerda:**

Terv: JS fundamental-ok kipróbálása, import/connect to database/geoCode/Json, dinamikus egységként való megjelenítés (Táblázat + térkép)

Haladás: A mai napra tervezett dolgok után kutattam, délután tartottunk egy megbeszélést ahol a tanár úr elmagyarázott 2 térképészeti optimalizációs módszert is. Utána néztünk hogy mi fog kelleni a projekthez: Szerver oldali Servic-ek, új adattábla, polygon struktúrák. Át kéne beszélni a front-endet is hogy mik azok az információk amiket meg szeretnénk jeleníteni, mert úgy könnyebben írhatunk kódot hogy hozzáférhessünk a kívánt adathalmaz-hoz.

**2019.07.18. Csütörtök:**

Terv: A Cserép Mátéval megbeszélt projekt alapjainak előkészítése: Angular, Emailben kapott sok-sok link tanulmányozása, dinamikus egységként való megjelenítés (Táblázat + térkép), tervezés, Service írás tanulmányozása

Haladás: A nap jelentős részében egy Angular alap projektet igyekeztünk összehozni. A terv az hogy 4 modullal fog rendelkezni, így ennek megfelelően elő is készíttettük a projektet majd feltöltöttük GitHub-ra. Bal felül egy statisztikai szekció lesz, az alatta (bal alul) megjelenített lekérdezett adatokból. Jobb oldalt a két típusú térkép fog megjelenni. Mivel külön modulok ez egyes elemek, így mindenki tud dolgozni külön modulban és még merge conflict-al se kell bajlódni. Még nem sikerült megoldani hogy az OpenLayer map működjön, de holnap ezzel kezdünk.

**2019.07.19. Péntek:**

Terv: Újabb megbeszélés Cserép Mátéval, haladás a projekttel.

Haladás: A nap javarészt kompatibilitási problémák megoldásával ment el. A reggeli megbeszélés után tisztáztuk a célokat, az alkalmazható eszközöket, és kaptunk adathalmazt amivel dolgozhatunk. Az első probléma az adatok feltöltése volt az adatbázisba. Ez a geoJSON sokoldalusága okozta, erre még keressük a megoldást, mert bár rájöttünk hogy a SAP HANA támogatja a mindenféle GIS-hez kapcsolódó adattípusok tárolását, még így is összetett az adat. Próbálkoztam megjeleníteni az adatokat a használandó könyvtárak egyikével (jelenleg Leaflet-ben), de sikertelenül. A probléma lehet a kapott adatok sűrűsége.

A minta térképeken lévő adat eléggé low poly, (azok megjelennek) de a kapott adatok gyanításom szerint túl sűrűek, ezért nem tudja megjeleníteni. Lefordul, eléggé laggy, de nem tölti be a saját adatainkat, csak a meglévő példákat.

**2019.07.22. Hétfő:**

Terv: A tanár úrtól kapott minták alapján megvalósítani a két térképet, aztán apró lépések: Megye fölé húzott egér: valamilyen statisztikát mutasson, megyére kattintva, kilistázza mondjuk a településeit, amiket contains() metódus al számolunk ki, megyék pl népesség sűrűsége alapján színezni.

Haladás: 3-an 3 félét kezdtünk el. Az adatok feltöltésével ismét elakadtunk, mert úgy néz ki hogy az egyetlen alkalmas adatbevitelre szolgáló módszerhez nincsen jogosutlsűgunk.

Ezen felül Angular-ral támadtak problémáink, ugyanis egyik könyvtárhoz sem vagyunk képesek a kapott példát SE működésre bírni. Nagyon lassan haladunk, nem találunk megoldásokat a problémáinkra. A példák önmagukban tökéletesen működnek, de angular-ban nem működnek, úgyhogy ha holnap a tanár úr nem tud megoldást rájuk akkor kiemeljük a projektete angular-ból.

**2019.07.23. Kedd:**

Terv: Konzultáció a tanár úrral, hiszen mind a 3 szálon elakadtunk a projekttel. Bebugolás, repó rendbetétele.

Haladás: A Tanár úrral tartott konzultáció során, hosszú órákon keresztül kompatibilitási problémákat próbálunk megoldani 4-en. A legfőbb probléma a proxy hiánya volt, hiszen a böngésző protokollok nem engedik, hogy azonos számítógépen/szerveren lévő adatot elérjen egy script. Ez úgy lett ideiglenesen megoldva, hogy a caesar oldalról php szolgálja ki a kliensnek a geojson-t. Ezzel meg lett kerülve a CORS policy probléma.

**2019.07.24. Szerda:**

Terv: A tegnapi megoldást alkalmazva az apróbb hibák elhárítása, majd azonos projektből mindenki a saját modulját fejleszti. Tervek: Kattinthatóvá tétel, valamilyen paraméteren keresztül a térkép színezése (pl: lakosság, települések száma)

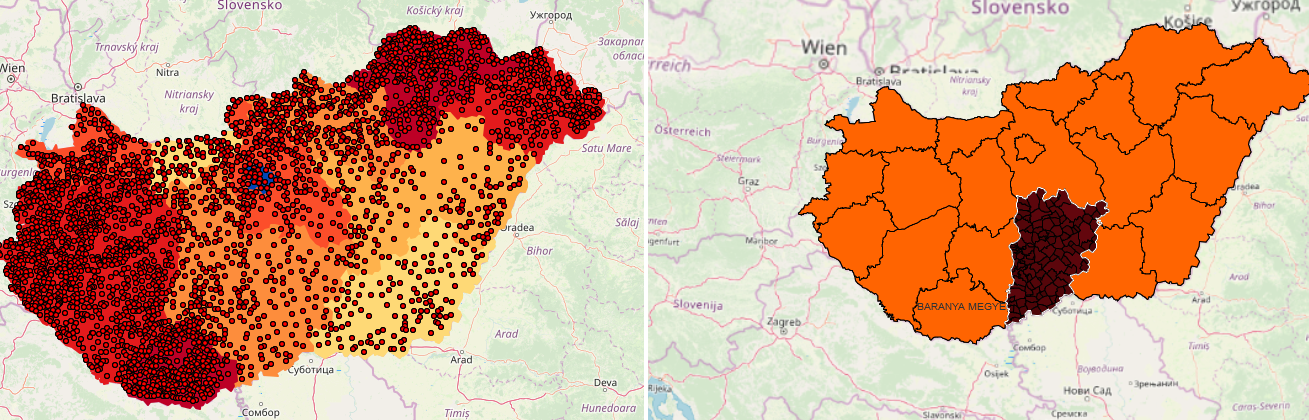
Haladás: Sajnos még mindig nincs se tanúsítvány a szerverhez, se jogosultságunk, hogy feltötlsük, a megfelelő objektumokat tartalmazó táblákat, hogy a szolgáltatásokat meg tudjuk írni, így maradtunk a közvetett geojson-ös beolvasásnál. A korábban a kódban hagyott hibás részletek miatt több időt vitt el a kód megtisztítása, rendbetétele, de a nap végére sikerült elérni, hogy a megyék mindkét térképes modulban színezhetőek, és kiválaszthatóak. Holnap ezeket a funkciókat tervezzük fejleszteni, véglegesíteni.

Terv a projekt végére: Összehasonlítani a leaflet-et, és az openlayers-t és leírni, hogy mennyire volt nehezebb az egyik a másiknál, milyen nehéz volt elérni ugyanazt az állapotot/funkcionalitást.

**2019.07.25. Csütörtök:**

Terv: Funkciók bővítése, back-end első felének kidolgozása.

Haladás: Hosszas szenvedés után sikerült elérni, hogy a megyék külön színezhetőek legyenek, továbbá sikerült betölteni és bug mentesíteni a városokat. További feladatok lesznek majd: A megyék kattinthatóvá tétele, a városok filter-elése, a megyék színezése és egyéb funkciók bővítése. A mai nap nagy része elment debugolással hála a böngészőknek.



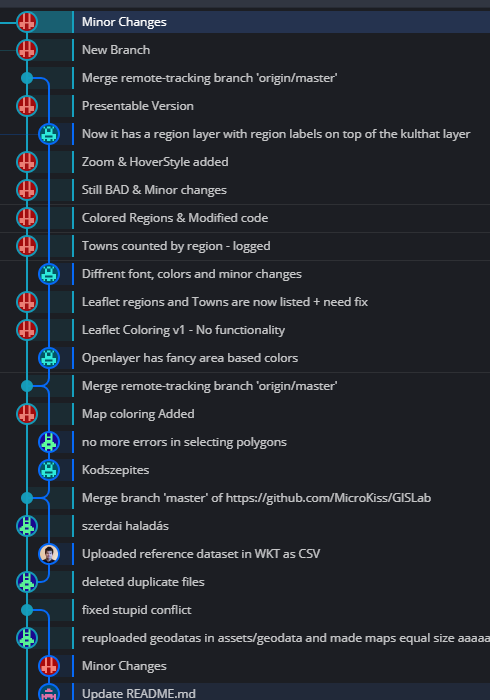
**2019.07.26. Péntek:**

Terv: Funkciók bővítése, back-end befejezése

Haladás:Sajnos nagyon sok idő elment a leaflet könyvtár butaságaival, hiszen, az adatok beolvasása után nem Marker típusú objektumokat jelenít meg, hoz létre hanem egy saját típust aminek nem állíthatóak a paraméterei. E miatt ma a térkép megyéinek a formázásával foglalatoskodtam többet. (A településekkel elment kb 2 óra szenvedés 0 haladással) A megyékkel sikerült elérni a fenti képen látható állapotot, továbbá annyi nem látszik, hogy működik a megyék kijelölése és rá nagyítása, és vissza színezése. Jelenleg probléma hogy a települések nem részei a megyék polygon-jainak így azokra húzva az egeret visszaváltozik a megye az eredeti színére. A jelenlegi verziók mergelve lettek a master branch-re és hibátlanul működnek. Nevezhetjük 0.5-ös verziónak (v0.5)

A mai nappal eltelt az első 4 hét, így a megbeszéltek alapján a következő 2 héten csak távolról, kevesebbe időt rá fordítva tudok dolgozni. Ennek a munkamenetnek szintén nyoma lesz a git repo-n hiszen továbbra is commitolni fogok.

A repo karban van tartva és rendeltetésszerűen használjuk:



**2019.07.29. Hétfő:**

Terv: Fejlesztési problémákra megoldást találni

Haladás: A nap nagy részében úton voltam, így főleg csak a problémám megoldásán gondolkodtam. A probléma: a 2 adathalmazt aszinkron módon töltöm be, így biztosan ki kell emelnem a városokat betöltő elemeket egy külön függvénybe, és úgy gondoltam, hogy elérem hogy csak a megfelelő városokat töltse csak be, így elérhető hogy ne legyen lagg-os az oldal a 3000 településtől, továbbá a kezelést is egyszerűbbé tenné. A következő probléma, hogy a towns let-ből nem tudom hogyan kell kiszedni a kívánt városokat, és csak azokat megjeleníteni.

**2019.07.31. Szerda:**

Terv: Korábbi megoldást implementálni, konzultálni a többiekkel a haladással kapcsolatban.

Haladás: Sikerült a kódot optimalizálni, átírni, továbbá elő lett készítve hogy service-el és anélkül is megírható a program. 2 Bug-ot sikerült megtalálni, és az új verzió által okozott error-okat, warning-okat megoldani. Még nem került kommitálásra.

**2019.08.02. Péntek:**

Terv: Megcsinálni service-k nélkül hogy a megfelelő városokat töltse be a kiválasztott megyénél.

Haladás: Elkészült! A Városok most már csak a kiválasztott megyében jelennek meg, így kevésbé laggos az oldal, mivel nem kell az összes települést megjeleníteni. Amennyiben a felhasználó másik megyét választ ki a korábbi városok eltűnnek.

**2019.08.05. Hétfő:**

Terv: Megcsinálni service-ekel a Pénteki állapotot egyeztetni a tanár úrral a további fejlesztési lehetőségekről, továbbá a Vincellér tanár úrral a pótlás milétéről.

Haladás: Projektet merge-elése, servicek implementálása, alkalmazása, együttes vizualizáció. MEGTERVEZTÜK hogy hogyan legyen a jövő heti munka, továbbá a 2 hét home office. A tervek: Megyék színezése a településben található cégek össz forgalma alapján (kész), településekre kattintva az ott található cégeket kilistázni, (csak azok a települések kattinthatóak ahol van cég). Minimális statisztika implementálása.

Későbbi terv: A települések kiválaszthatóak legyenek, és adatokat kiírni pl pin-ként.

**2019.08.06. Kedd:**

Terv: BUG-ok keresése, javítása

Haladás: Volt 3 nagyon csúnya bug a kódban, egy jó 2 óra elment a log-olgatással, aztán kiderült hogy a fő errorokat dobáló probléma nem a leafletben volt hanem az openlayersben.

Zavaró log-okat is eltávolítottam, mert el ömlesztették a konzolt.

Az openlayers-ben egy console.log() ki-kommentelése során keletkezett egy furcsa bug, azt is megoldottam.

**2019.08.07. Szerda:**

Terv: Térkép fejlesztése, service-k kérése Jánostól, azok implementálása, debuggolás.

Haladás: A megjelenített pontok mostmár Marker-ek, ez azért fontos mert egyedi műveleteket tudunk hozzájuk rendelni, így pl popup-okat megjeleníteni, mondjuk a város nevét.

Megvalósítottam hogy a megykére való kattintáskor a szervertől kérdezze le a megyében szereplő városokat, és írja ki az adatokat.

Hosszas keresgélés után egy hiányzó zárójelet is megtaláltam, majd formáztam a kódot.

**2019.08.08. Csütörtök:**

Terv: Térkép fejlesztése, újabb, szükséges service-k kérése Jánostól.

Haladás: Nagy előrelépés: Sikerült működésre bírni a városokat megjelenítő szolgáltatást, így a megyékre kattintva azok a városok amikben található az adatbázisban cég, azokat zöldként a többit piros pont ként jeleníti meg a program.

**2019.08.12. Hétfő:**

Terv: Error javítása: valami baj van az “ő” betűvel.

Haladás: Kiderült hogy az escape az ő betűket rosszul konvertálja, ezért le lett cserélve kliens, és szerver oldalon is encodeURI-ra, így 2 error is megszűnt.

**2019.08.13. Kedd:**

Terv: Zöld pont, javítása, statisztika megjelenítése,

Haladás: Zöld pont most már transzparens. Alapértelmezett statisztikának Pánd városa be lett állítva, és minden zöld település kattintható, továbbá kattintásra megjelennek a településen található cégek nevei, vagyonai, grafikonban. (TODO: (Server side): Rövid név jelenjen meg)

**2019.08.29. Csütörtök:**

Dokumentáció megtervezése, Beszámoló metrevezése

**2019.08.30. Péntek:**

Dokumentáció megtervezése, csoport többi tagjával való kommunikáció

**2019.08.31. Szombat:**

Kód átnézése, refaktorálása, Beszámoló megírása