HungaryVFR X-plane Scenery Developer Team  
**GITHUB Használati útmutató**

1. Regisztráció a Github-ra, kliens telepítése

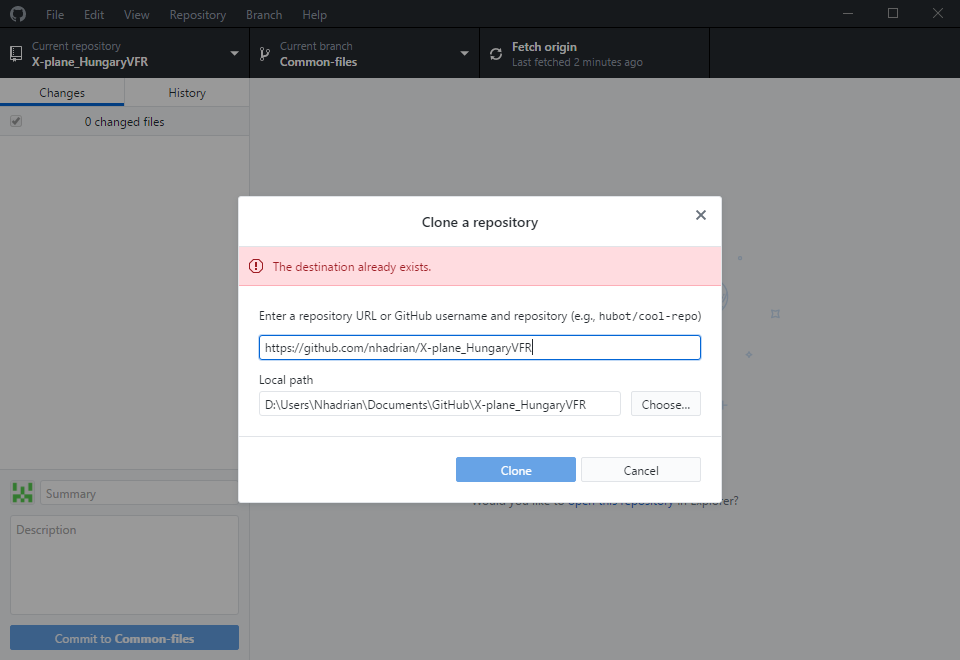
* Regisztráció (ingyenes): https://github.com/
* Github kliens letöltése és telepítése: <https://desktop.github.com/>

A program telepítését követően meg kell adni az alapértelmezett könyvtárat ahova a Github kliens klónozni fogja a tartalmakat, a jövőben az alapértelmezett könyvtárban lévő file-ok lesznek feltöltve a Gihub-ra. **NEM SZABAD** az X-plane Scenery könyvtárat megadni mert akkor mindent ami abban van klónozni fog, pl. library-ket, egyéb scenery-ket!!!

* Kliens elindítása és belépés a felhasználói adatokkal.

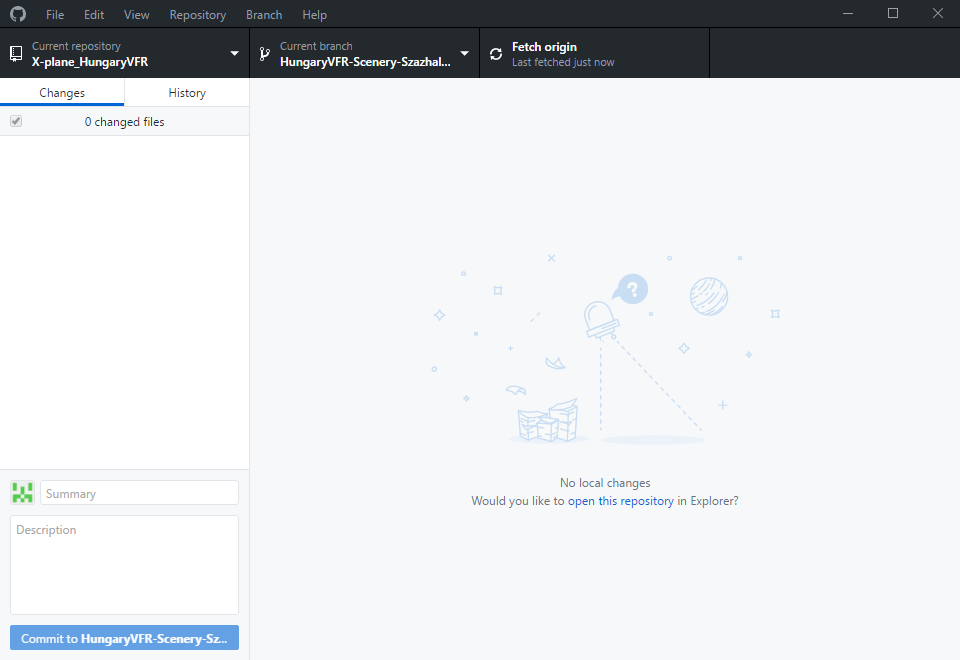
**FONTOS!!! A Github-on és a file nevekben lehetőleg soha ne használjunk ékezetes betűket és karaktereket hogy a platformok (pl. PC, MAC, Linux) közötti kompatibilitás a lehető legjobb legyen!**

* A kezdéshez klónozni kell az X-plane\_HungaryVFR repository-t a saját gépre. Ezt a file/Clone repository menüpontban lehet megtenni, itt be kell illeszteni az alábbi linket:[**https://github.com/nhadrian/X-plane\_HungaryVFR**](https://github.com/nhadrian/X-plane_HungaryVFR)



1. Github file struktúra, kliens felépítése

* A Github-on egy repository-n belül branch-ek vannak. Ez azt jelenti hogy lehetőség van a nagy projekt (HungaryVFR) egy részén (branch-en) dolgozni anélkül hogy az egész adatállományt befolyásolnánk. Ez azért jó mert így például két-három fejlesztő tud külön egy scenery-n is dolgozni ha éppen úgy alakul a fejlesztés irányvonala. A branch-ek csak a hozzá tartozó file-okat kezelik, erről majd még később részletesen írok. **A master branch-ba bármit menteni tilos,** azt szeretném megtartani üresként az újonnan létrehozandó scenery-k sablonjaként. Helyette használjuk a ’HungaryVFR’ nevű branch-et a közös adatállománynak.
* **FONTOS!** A Github kliens nem mutatja az adott brench-ben található összes file-t, csak mindig a változásokat. Ezért a file listához meg kell nyitni file kezelőben a Github mappát és a kliensben ki kell választani a megfelelő branch-et. Később még lesz róla szó.

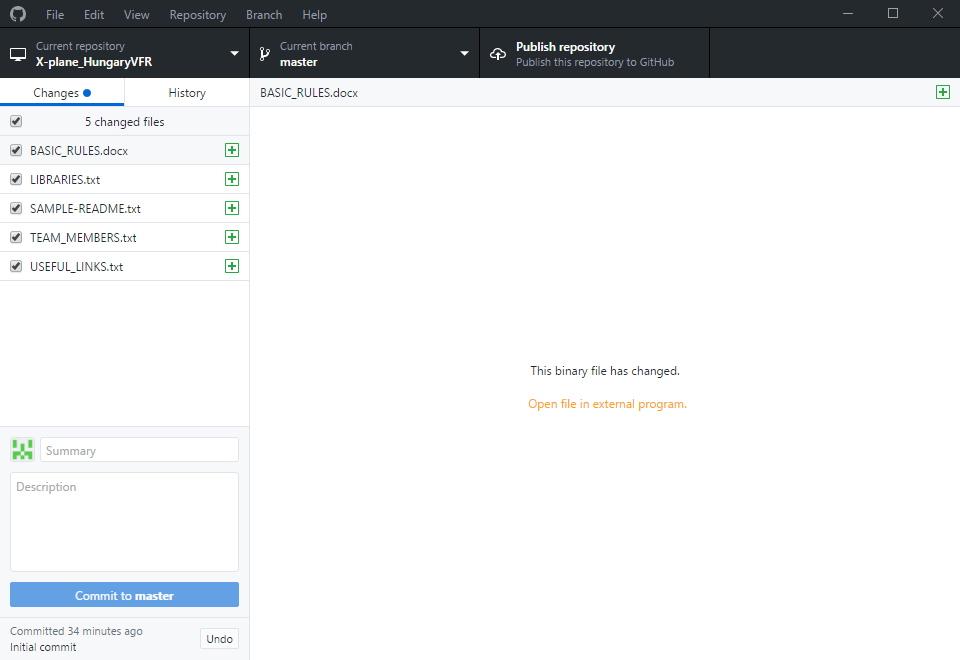


**Current branch:** aktuális branch kijelzése valamint branch választó  
**Fetch origin / Publish Branch:** le és feltöltés gombja  
**Changes/History:** Megmutatja a változásokat valamint az előzményeket  
**Summary/Description/Commit to…:** itt lehet kitölteni a változások verzióját és kommenteket majd hozzáadni a branch-hez a változásokat (Figyelem, a hozzáadással még nem kerül online feltöltésre, lást később!!!)

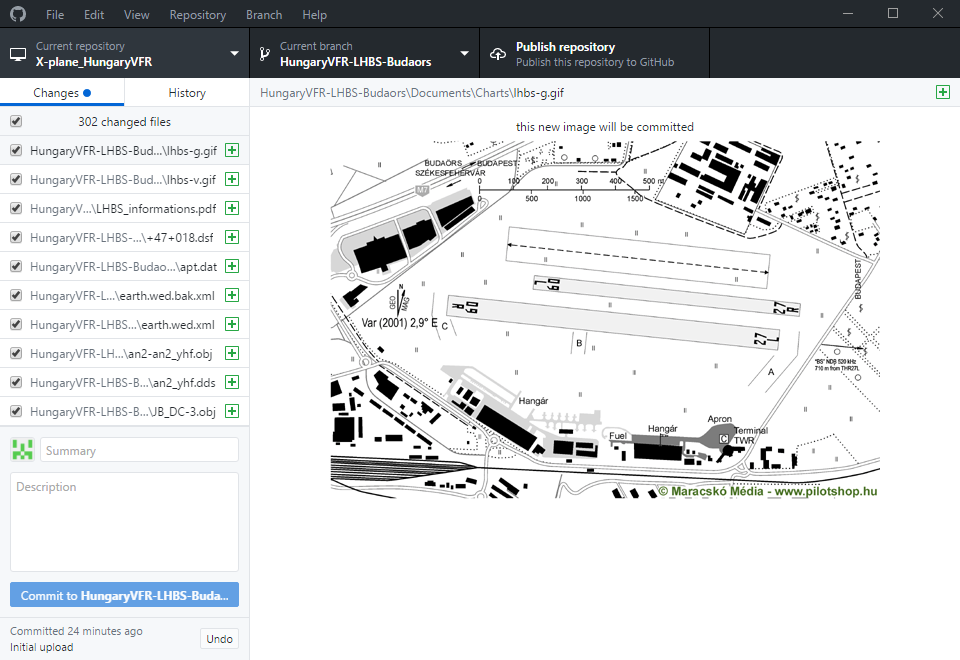
1. Új branch hozzáadása (új scenery feltöltése)

A Githubra a változtatásokat úgynevezett „Branch”-okban javaslom közzé tenni, ez azért is előnyös mert egy közös adatbázist többféle irány szerint lehet fejleszteni, egymással párhuzamosan. Én javaslom hogy scenery-nként készítsünk egy Branch-et és azon belül dolgozzunk, így egymástól függetlenül is kipróbálhatóak egy-egy scenery különböző változatai, valamint bármikor befrissíthető a Master branch-be.

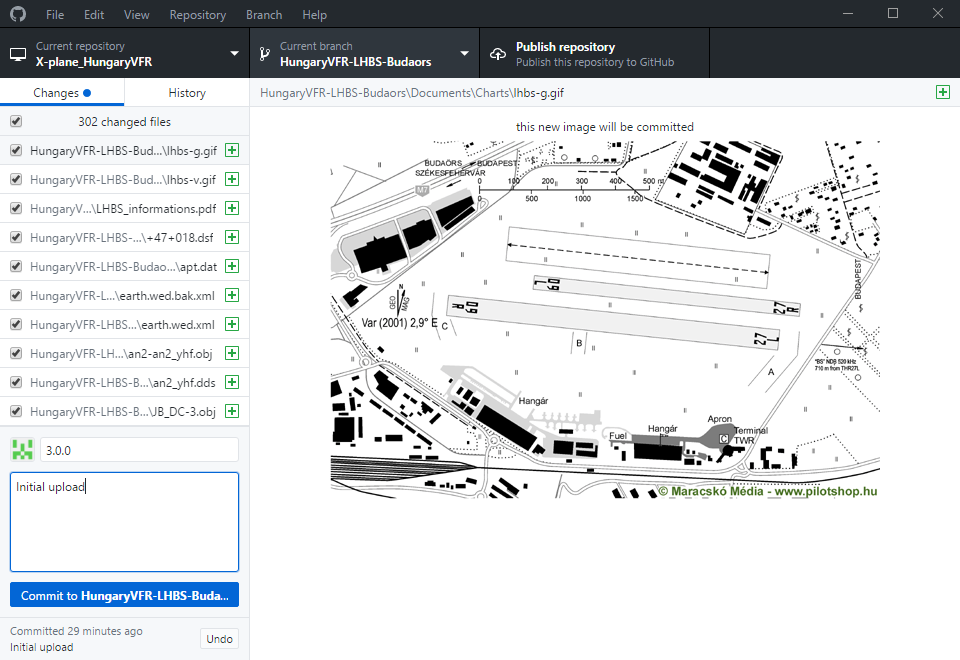
* Első lépésként másoljuk be a megosztani kívánt tartalmat a Github/X-plane\_HungaryVFR/ mappába. Ezt követően indítsuk el a Github klienst. Egyszerre csak egy scenery-t töltsünk fel!!! *(Ha a fejlesztés során több scenery-hez is hozzányúlunk akkor csak scenery-nként másoljuk be a Github mappába és töltsük fel a kliensel!!!)* A kliensben meg fognak jelenni a változtatott file-ok:



* Ha új scenery-t töltünk fel akkor létre kell hozni egy Branch-et, amit a Branch/New Branch menüpontból lehet megrenni. Példaként létrehoztam az LHBS scenery branch-ét:



* Feltöltés előtt mindig ellenőrizzük, hogy a megfelelő Branch-ben vagyunk!!!
* Ezt követően adjuk meg a változtatás verziószámát valamint a leírásban röviden részletezzük a változásokat, hogy a többi fejlesztő lássa mi történt. Ha próbára szeretnénk valamit feltölteni akkor a verziószám mögé írjunk ’Beta’-t!!! A verziószám minden esetben egyezzen meg a mellékelt README.txt-ben szereplő verzió számmal, kivéve ha Beta változatról van szó!!! A verziószámok képzéséhez a Basic Rules-ban találtok támpontokat.



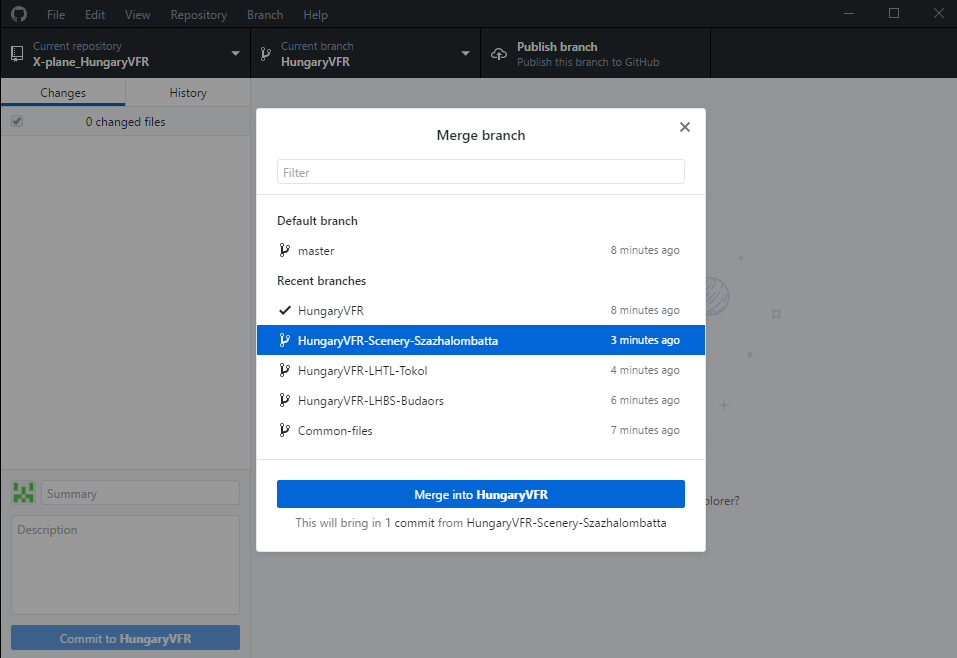
* Ezután nyomjuk meg a ’Commit to HungaryVFR… ’ gombot a mentéshez, majd a ’Publish branch’-t az online feltöltéshez.
* **FONTOS!!!** Ha fut a kliens és branch-et váltunk akkor a Github mappán belül mindig csak az adott branch file-jai lesznek láthatóak! Ezért érdemes először a kliensben branch-et választani vagy váltani, ezt követően felülírni a mappa tartalmát a módosított scenery-vel majd utána feltölteni.

1. Meglévő branch módosítása

* **FONTOS!!!** Mielőtt szeretnénk egy adott Scenery-ben módosításokat végezni, győződjünk meg róla hogy a legfrissebb verzióval rendelkezünk mert elképzelhető hogy idő közben valaki már változtatott rajta. Ezért indítsuk el a klienst, válasszuk ki a branch-et majd nyomjunk rá a ’Fetch origin’ gombra az ellenőrzéshez, ezt követően a github mappából másoljuk a Scenery tartalmát az X-Plane-be és úgy kezdjünk el fejleszteni.
* Mint írtam korábban, a Github mappában mindig csak az aktuális branch látható. Ezért feltöltéshez először inditsuk el a klienst, válasszuk ki a módosítani kívánt branch-et majd írjuk felül a mappában található Scenery-t a módosított file-okkal. Egyből meg fognak jelenni a változások. A verzió módosítás adatait kitöltve ismét a ’Commit to ….’ gombra kell kattintani, majd a ’Push origin’-re.

1. Branch összevonása a **’HungaryVFR’**-el

* Mivel a ’HungaryVFR’ branch tartalmazza kompletten az egész fejlesztést, ezért egy-egy végleges módosítás után (ha a verzió szám már nem tartalmazza a Beta kifejezést) az adott Branch-et össze kell vonni a HungaryVFR-el. Ehhez először válasszuk ki a ’HungaryVFR’ branch-et majd a Branch menüpontból a ’Merge into current branch’ pontot. Itt ki kell választani hogy melyiket szeretnénk a HungaryVFR-be bemásolni, majd rá kell nyomni a „Merge into HungaryVFR”-gombra.



Sikeres fejlesztést!!! ☺