This tutorial supposes next

1. You have basic experience in BattleTech modding, eg. you can create mod.json, know about mods manifests etc.

2. You have basic experience on Unity editor eg. you know how to manipulate items, add scripts to objects, know about MeshRenderer, MeshFilter, Terrain, etc concepts.

3. You know how to use Notepad++ and similar text editors, you know how to copy files, know how to understand what assembly version is.

Software pre-requirements

1. BattleTech with latest RogueTech installed. Potentially it can work even with vanilla but i've developed it for RogueTech, debugged in RogueTech environment not tested in other environments. If you are modpack maintainer and going to respect my authorship, you can contact me (KMiSSioN) at RogueTech discord (https://discord.gg/93kxWQZ) to discuss permissions and solve compatibility problems.

2. Unity Editor version 2018.4.f3-11 (game itself using Unity engine version 2018.4.f3). I'm using 2018.4.f11. You should not experience any problem if using any version in between.

3. Custom version of AssetRipper. Original version unfortunately have one problem – ones it encounters asses it can't export it throws exception and stops. My custom version skips assets it can't export and proceed. There are three possible sources of mentioned tool.

    a) Get sources from my fork repository and build them by yourself. VisualStudio 2022 required. Fork https://github.com/CMiSSioN/AssetRipper

    b) Get specific AssetRipper version binaries (2.4.1 - unfortunately 2.4.2 have worse performance) from original repository and replace AssetRipper.Fundamentals.dll to one I'm providing https://github.com/CMiSSioN/CustomMaps/blob/master/CustomMaps/Tools/AssetRipper/AssetRipper.Fundamentals.dll

    c) get whole AssetRipper build by me from my fork of original repository. Here https://github.com/CMiSSioN/CustomMaps/releases under "Assets" you you will found some assemblies and tools you need.

4. Latest versions of CustomMaps and CustomMapsEditor assemblies. Here https://github.com/CMiSSioN/CustomMaps/releases under "Assets" you will found some assemblies and tools you need.
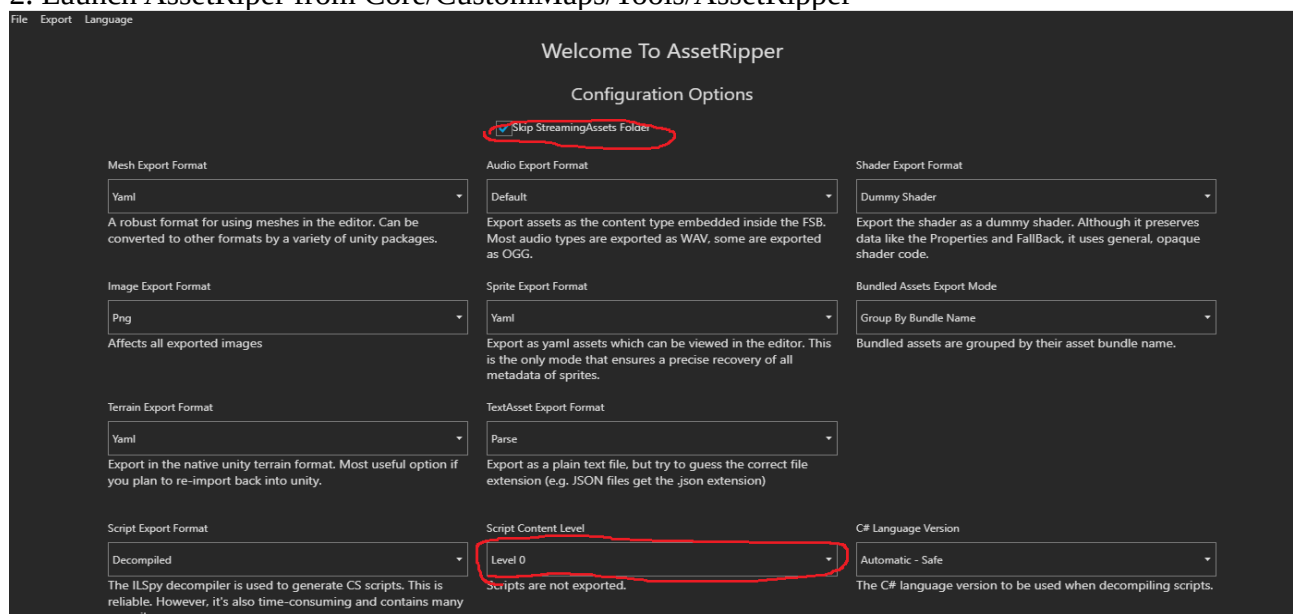
Preparation.

1. Close all "heavy" application, cause exporting process needs as many resources as your PC can provide.

2. Unpack file from https://github.com/CMiSSioN/CustomMaps/releases to you Mods/Core/CustomMaps folder with replace. Your CustomMaps folder should looks similar to this


if you have CustomMaps subfolder in your Mods/Core/CustomMaps – you've done it wrong.

2. Launch AssetRiper from Core/CustomMaps/Tools/AssetRipper

and set options as on screen.

3. In main menu choose File/Open Folder and select BattleTech_Data folder. Prepare to wait. Once interface show you assets tree in main menu choose Export/Export all files. Choose destination folder with as short path as possible. Prepare to long wait.

4. Close AssetRipper. Navigate folder your choose for an export find folder where Assets sub-folder is. Copy all folders from Core/CustomMaps/Tools/ForProject to this folder with replacement. Also delete Assets/Scripts folder in your unity project.

5. Open unity editor and open project (choose folder you've found on previous step). Prepare to wait even longer than while export.

6. Once interface is shown close Unity Editor

7. Open RT launcher, disable safe launch and start game. **It is better to disable extending map boundaries and randomizing spawn positions for both skirmish and career in MissionControl settings**

8. Go to the skirmish (in this tutorial i'm targeting skirmish maps cause it is faster to test, once you get basics you can try modify career map, process is similar)

9. Start River Crossing map (for example). **On briefing screen before launch contract but when launch button is active** press RightShift + F1. You will get a prompt for name for new map.
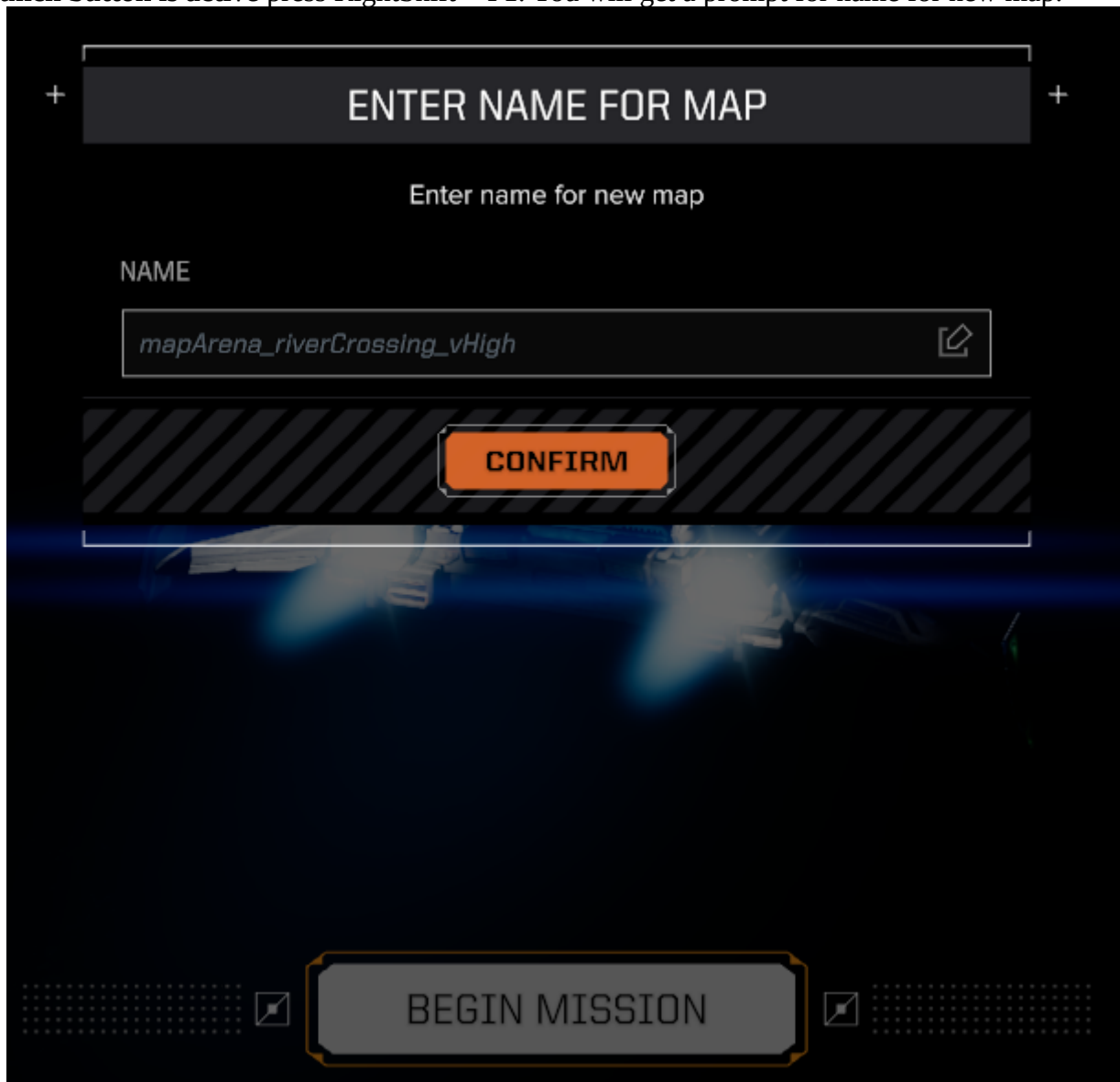
Enter name "mapArena_riverCrossing_vHigh_test" (for example). Name can be any, but without spaces and special characters other is on your consideration. It will save your current loaded map runtime info to Mods/Core/CustomMaps/<name you've entered>. **If folder is exists it will delete it before export.** Now close game and prepare yourself to edit your first map.

There is video can help you in process https://youtu.be/z3nvopOsjyk

Map editing.

1. Open unity editor and open project you've create on stage 5 of preparation. It should be in menu now so you had not to open folder.

2. In Assets/Level Design/Maps open scene with map you are about to edit (and exported at stage 9 of preparation). If you follow instructions it is mapArena_riverCrossing_vHigh.

3. Save with different name. mapArena_riverCrossing_vHigh_test for example. Good idea to have the same name as you've entered at stage 9 of preparation. It is needed to keep original scene intact.

4. Go to the <Unity project folder>/Assets/TerrainData with OS explorer or similar tool, find mapArena_riverCrossing_vHigh_terrain_x0_y0.asset file and make two copies of if. Rename first

copy as mapArena_riverCrossing_vHigh_test_terrain_x0_y0.asset second as
mapArena_riverCrossing_vHigh_test_metadata_x0_y0.asset. It is needed to keep original terrain data intact.

5. Open terrain data copies you created with text editor and change m_Name field for both copies accordingly

6. Return to Unity Editor. Select terrain. Switch inspector to debug mode and change "Terrain Data" to your copy (mapArena_riverCrossing_vHigh_test_terrain_x0_y0) for Terrain and TerrainCollider.

7. Duplicate scene terrain name it something like metadata_instance_x0_y0 (actual name is on your but you need to know where is Terrain for terrain and where is Terrain for metadata). Change "Terrain Data" for metadata Terrain to mapArena_riverCrossing_vHigh_test_metadata_x0_y0 (again both Terrain and TerrainCollider instances). Switch inspector to normal mode.
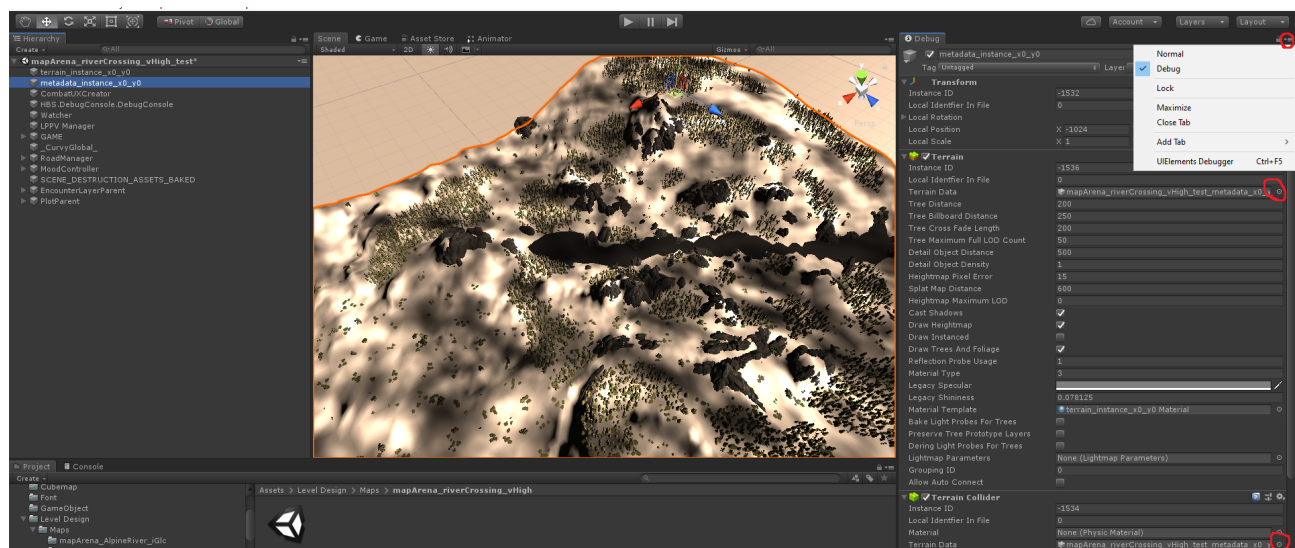
8.Delete all layers from both main terrain and meta terrain, delete all trees from metaterrain, change terrain material to standard both main and meta terrains, add main terrain TerrainUpdateTracker script, create new empty object, name it MapImporter (or whatever you want) and add it MapImporter script, assign terrain, meta terrain to its fields, set its MapDirectory field (it is path CustomMaps mod saves your map data on stage 9 of preparation, Mods/Core/CustomMaps/<name you've entered> without backslash). Select MapImporter press "Delete Missing Scripts" button, save scene, close unity editor.

9. Open unity editor again. Delete unnecessary objects. Adjust scene light. Select MapImporter and press CreateLayers, wait for finish. Press ImportMetadata, wait for finish. Press ImportTerrainData, wait for finish. Press PlaceMetaMarks, wait for finish. Press ImportObjectsData, wait for finish.

10. Now you can edit your terrain. At this stage it is better to disable meta terrain to not interfere, you will handle it later. What you can do (if this tutorial does not say you can do something directly, you are certainly can't unless you know what are you doing).

a) you can change terrain height

b) you can add or delete terrain trees from existing pallet

c) you can move/scale/disable certain objects. Certain is 1) having GAMEItemComponent script and 2) having movable flag (note this flag is only marker, you can set it on by yourself but object will not gain ability to be moved. It is caused HBS made most rocks on map one big mesh, individual rock refer submesh of this big mesh)

objects without IsMovable flags can be only enabled/disabled.

d) you can duplicate existing objects (Duplicate button on GAMEItemComponent script)

e) you can add new items if you have prefab for them

f) you can move encounter logic objects (EncounterObjectGameLogic). They represent units spawn points, encounter bounds etc. You might notice black cube covering part of map – it is an encounter bounds.

g) you can paint layers from existing pallet over terrain

h) you can change metadata (say to the engine which part of map have forest, water, road, etc)

to duplicate existing GAMEItemComponent select it and press "Duplicate" button. Do not duplicate map item other than using "Duplicate" button unless you are certain sure what are you doing. Do not move your new items in hierarchy – main rule new items should always go after existing ones.

to add own object to map – duplicate existing object (preferably from Parent_Structures), remove all of its inside, set your prefab (it should be presistant object) and press "SpawnPrefab". If you want your new object have impact on metamap you should press "AddObstruction" (your object should have any collider and does not have ObstructionGameObject script on it, if it has ObstructionGameObject it means it have influence on metamap already).

Also you should note if you spawning not duplicated object – standard shader might not display correctly in game, that it why you need set existing material source (Material field). Object can be material source if has two scripts any Renderer (MeshRenderer, SkinedMeshRenderer etc) and SearchByNameIndex so it can be found in game. Replace textures flag controls if textures will be get from prefab material or renderer material you have selected.

Once you've finished with your main terrain modifications you should alter metadata terrain accordingly. First enable metadata terrain. Than select MapImporter and press SyncMetaTerrainHeghts this will sync your metadata terrain and main terrain heights maps.
Than you can disable drawing of the terrain surface for metadata terrain to see your main terrain surface while drawing over meta terrain. Colored cubes will help you to track actual metadata terrain state.
Sync your alterations to metadata terrain by painting tools.

Once you've finished it is time to export your modified map. Press "ExportMetaData", "ExportTerrainData", "ExportRaycastData" and "ExportGameObjects" in order waiting it finished before pressing next button every time. ExportRaycastData may work long.

If you've added your own prefabs to map (i've added prefab Rock_01) you should build assets bundle and modify your map mod.json manifest accordingly. I've added Rock_01 prefab to river_crossing_rocks assets bundle.
To export your prefabs press "BuildAssetsBundles" in MapImporter script. This will build projects bundles. Than press "CopyAssetsBundles" this will copy built bundles to map folder.

In Mods/Core/CustomMaps/<your map name> if you followed this tutorial instruction it is Mods/Core/CustomMaps/mapArena_riverCrossing_vHigh_test there are two files you need to edit
1.Mods/Core/CustomMaps/mapArena_riverCrossing_vHigh_test/mod.json
2.Mods/Core/CustomMaps/mapArena_riverCrossing_vHigh_test/custom_maps_defs/
mapArena_riverCrossing_vHigh_test.json

and one you can use as source
Mods/Core/CustomMaps/mapArena_riverCrossing_vHigh_test/manifest.txt

in mod.json change "Enabled": false to "Enabled": true and add content of manifest.txt to Manifest section.
In mapArena_riverCrossing_vHigh_test.json – set proper FriendlyName
Also you can add icon to your map, just place relevant mapArena_riverCrossing_vHigh_test_icon.png to Mods/Core/CustomMaps/mapArena_riverCrossing_vHigh_test/map_icons folder.

Time to test your modified map. Go to skirmish if all done correctly you will see FriendlyName of your map in dropdown list. Select it, start map and you will see your changes been applied.

For career maps process is similar. Only difference career maps have more than one encounter layer. Certain encounter layer can be activated by pressing "EnableLayer" button in EncounterLayerData script.

Encounter layer it is roughly speaking game objects for certain contract type on this map. They a becomes active only if certain contract type loading, otherwise they a disabled.

Unity assets credits:

Double Sided Standard Diffuse Bump shader is part of "Free Double Sided Shaders" by Ciconia Studio

Rocks prefabs are part of "Rocks HD Pack" by ROMPAS studios