Getting satellite images for your terrain

Preliminar work: (don't do again if you already did it)

1) Create your .geo file (join_geos, for example), process it and save the mesh. Open the mesh with gmsh and check that the non-driveable zone (Physical Surface 222) is part of the mesh.

2) Run trocea_malla (inside s1_mesh)

Now we use SasPlanet to get the images

3) Create the .hlg file:

```
> cd s1_mesh
> create_hlg
```

A file called s1_mesh\grid.hlg has been created with the coordinates of a box that comprises your terrain.

4) Open s1_mesh\grid.hlg with **SASPlanet**:

Operations -> Select -> Load from file -> Load -> Start

5) Save the images in a folder (c:\example_folder) with the desired splitting (5x3 in this example):

Operations -> Select -> Previous Selection -> Stick -> Start (YOU **MUST select .dat file bindings)**

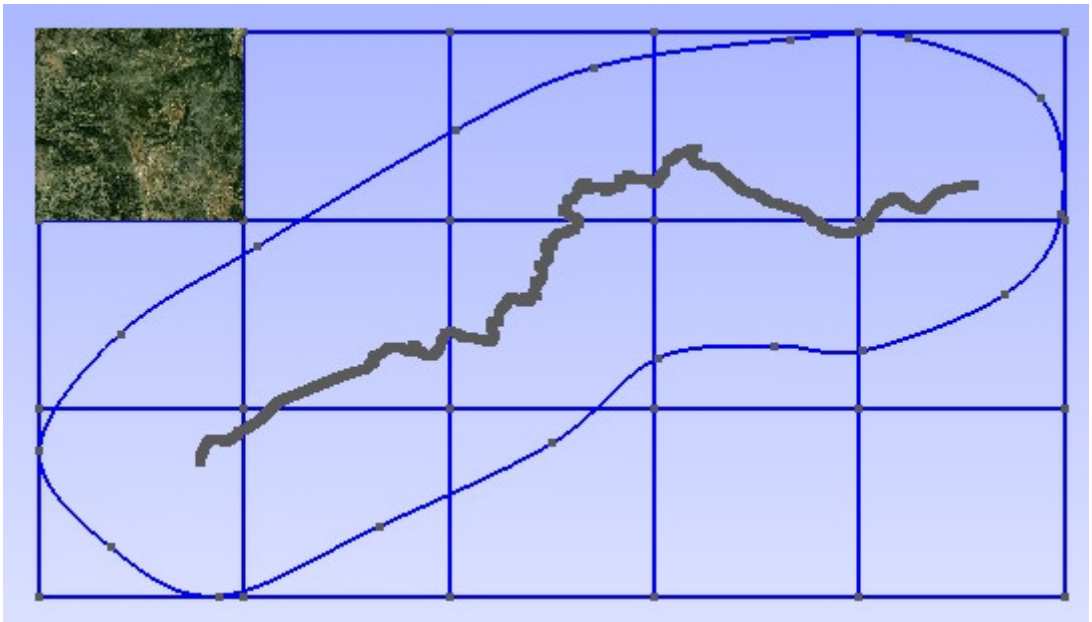
The satellite images have been splitted and the info about the splitting coordinates is in the created .dat files.

6) Now we can add the grid to joined.geo so we can see the limits of the downloaded images:

```
> add_dat_to_geo('c:\example_folder')
```

(if you don't want to add the grid to joined.geo, just select the option "Create grid.geo" when asked)

7) Open joined.geo and you will see the limits of the downloaded images you have inside c:\example_folder



Using the new grid for simplifying texture mapping

If you use the lines of the new grid for splitting the non-driveable zone, may be you can then map the textures easily. So now you can decide to change the non-driveable zone.

Script simplificar now creates a separate .ply file for each surface belonging to the non-driveable zone, so they can be easily simplified. **All the files on folder salida\nc_splitted should be processed with MeshLab**, for simplifying or just to remove unreferenced vertex. When you have processed all of them, open s4_terrain\salida\nc_splitted*.ply with MeshLab (you can open all in the same operation)
View-> Show Layer Dialog

Then **Filters->Layer and ... -> Flatten visible layers**. This combines all the grid cell's meshes. Save result as salida\n.ply

NOTE: add_dat_to_geo also creates a list of all the background images, with the format needed by BTB. It is called **s1_mesh\list_bi.txt**. It must be inserted by hand inside the Venue.xml (really easy to replace the empty list of background images in the 3rd line of this file).

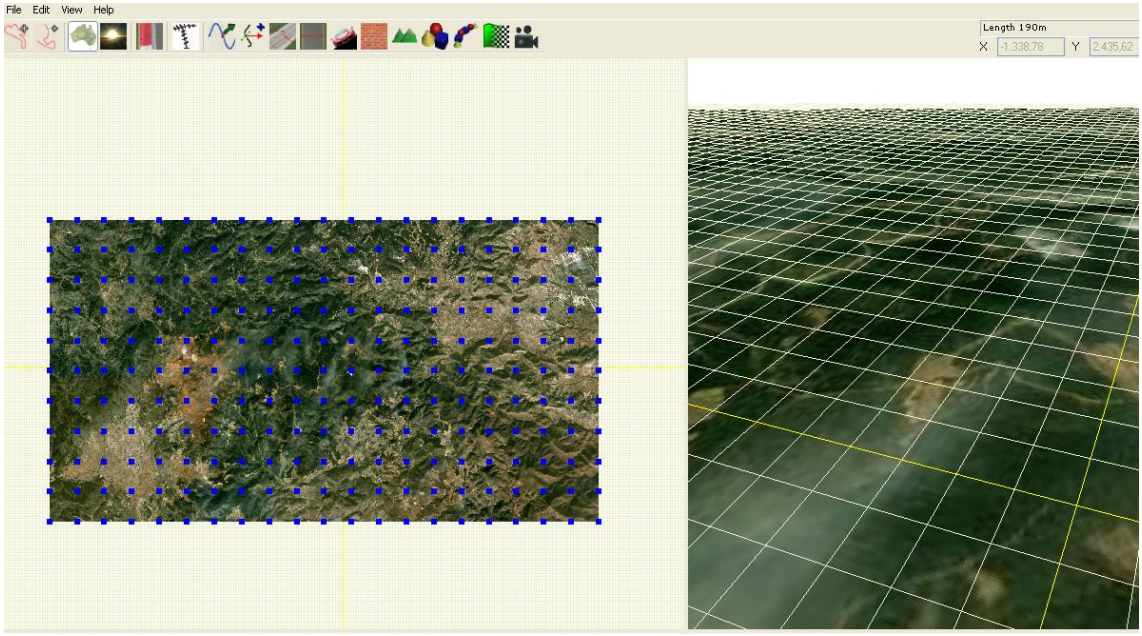
Images should be named sat_X-Y.dds and should be located inside
My Project\XPacks\Common\Textures
(*My project* is your project's BTB folder)

procesar_elementstxt_mt can automatically use the background images for the non-driveable zone, if the terrain is splitted using a grid of the same size as that used to create list_bi.txt.

For example:

procesar_elementstxt_mt(10,5,0) splits the terrain using 10x5 grid, but does NOT map background images to terrain zones. Inserting list_bi.txt and copying the images to Common\Textures folder will add the background images, but they will not be linked to the terrain.

procesar_elementstxt_mt(10,5,**1**) splits the terrain using 10x5 grid, using background images for blending in terrain zones. Before opening the Venue.xml It is mandatory inserting list_bi.txt into the Venue.xml and copying the images to Common\Textures folder. Otherwise BTB will refuse to open your project.



Example list_bi.txt

```
<BackgroundImages count="50">
    <BackgroundImage>
      <Path>Common\Textures\sat_1-1.dds</Path>
      <Plane>
        <Position x="-11232.375390" y="-0.5" z="5636.028398" />
        <Scale x="2100.341106" y="1" z="-2311.287859" />
        <Rotation x="0" y="0" z="0" />
      </Plane>
    </BackgroundImage>
    <BackgroundImage>
      <Path>Common\Textures\sat_1-2.dds</Path>
      <Plane>
        <Position x="-11232.375390" y="-0.5" z="3324.740539" />
        <Scale x="2100.341106" y="1" z="-2311.287859" />
        <Rotation x="0" y="0" z="0" />
      </Plane>
    </BackgroundImage>

etc.

  </BackgroundImages>
```

Summary

To get the images with SASPlanet you need a .hlg file with the info of your mesh boundaries.

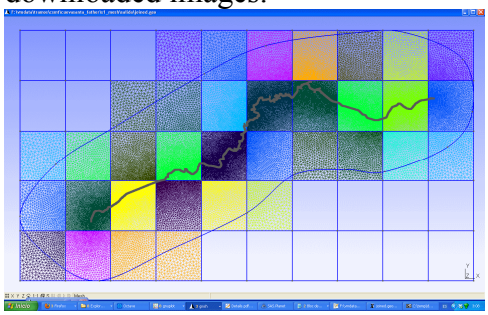
Script **create_hlg** makes that task: reads the mesh info (created from your .geo, saved as .msh and splitted with trocea_malla) and writes grid.hlg

Then you open grid.hlg with SASPlanet, following the steps in the doc, and you get the .jpg files. For each .jpg image a .dat file has to be saved. This .dat files contain the terrestrial coordinates of the image limits. Reading the .dat files with a script we can translate terrestrial coordinates to BTB coordinates and create a file ready to be inserted in the Venue.xml so the images are automatically placed in the background of the track. This script is: **add_dat_to_geo**. The output file is list_bi.txt. This file can be inserted into the Venue.xml and you will get your images as background.

That is a simple way to proceed. Now suppose you want to complicate things: you want a project with a terrain already configured to make blending between a default texture and the background images (If that is the case, you are sure you have created list_bi.txt and you know you have to insert it into the Venue.xml). Ok then, it is really easy: at the finals steps of the process you just have to call procesar_elementstxt_mt with 3 parameters: dimensiones of the matrix of images, and "1", to tell the script you want blending with background images:

procesar_elementstxt_mt(10,5,1)

But you want to complicate things even further: you want to have terrain areas that fit exactly your background images, as I did with my Cueva Santa track. Ok, if that is the case, then first you need to insert the limits of your images into the joined.geo, the .geo you are working with. it is really simple: when you run add_dat_to_geo, you have to select the option "add the grid to joined.geo". Then when you open joined.geo with gmsh, you will see the grid with the limits of the images you downloaded with SASPlanet. You have to do a little work merging those new lines with your existing terrain, creating smaller surfaces for the non-driveable zone. It is not difficult to do this step if you have a little practice using gmsh. In the image it can be seen the original mesh limits (the biggest spline curve), and the new meshes fitting the limits of the downloaded images.



Once you have your new mesh, you go on with the process as usual.

NOTE: I use Irfanview to batch resize .jpg with 2 power size, and DDSConvert to batch convert them to .dds.