

Tile Making

- an introduction

This guide is meant as an introduction to the Random New World (RNW) and the tiles that populate it aswell as a help for anyone who want to make their own tiles.

Personally i picked up map making to create tiles for the RNW and this guide is mainly aimed to help anyone trying to do just that. Therefore i have gathered the knowledge i have on how tiles work and described what i consider to be the best way of making a tile. Just remember there is no correct way of doing it aslong as the end result is a working tile.

-Poh

Index

Location of files related to RNW.....	1
How the RNW work, size, etc.....	2
How to make a tile.....	2
The text file.....	3
The Height map.....	5
The River map.....	6
The province map.....	7
Finishing Touch.....	8

Its assumed that GIMP version 2.8.14 is used for image manipulation.

Source for more information on map modding: www.eu4wiki.com/Map_modding

Location of files related to RNW

The files related to the RNW can be found in **[EUIVfolder]/maps/random**. Here you will find the files **RandomLakeNames.txt**, **RandomLandNames.txt**, **RandomSeaNames.txt** which contain all the random names lakes, land and sea provinces can get.

You will also find **RNWScenarios.txt** which contain information on the different nations which can inhabit the RNW.

This guide will not focus further on these but instead on the folder "**tiles**" which is where the fun begins. Inside the folder you will find numerous .txt files along with the subfolder "**data**" which contains .bmp files.

Every tile consists of one .txt file along with three .bmp files, a height, river and province map.

Now that this is out of the way its time to take a look at how the RNW work.

How the RNW work, size, etc.

The RNW can be seen as a large grid of squares which each measure 128 x 128 pixels. Any given tile takes up a number of squares depending on their size. The game chooses which tiles to use, place them into the grid and then generate seazones to fill the empty space between the tiles. It isn't completely random which tiles are chosen as it is affected by different factors which will be described under the .txt file.

The RNW measure in total **2304 x 2048 pixels** or the equivalent of a **18 x 16 grid**.

A tile can be anywhere from 128 x 128 up till 2304 x 2048 pixels. Since it is a grid the tile needs to operate in whole squares. So 128 x 256, 512 x 512, 2048 x 384 etc. Are examples of possible tile sizes.

Now the system has been defined its time to take a look at a single tile.

How to make a tile

As previously described a tile consists of four files, one .txt and three .bmp. The naming and function of the files is as follows:

1. Text file, **[name].txt**, containing the information regarding seazones, wastelands, lakes, estuaries, number of provinces, regions, tile size etc.
2. Height map, **[name]_h.bmp**, defining which parts of the tile is above and below water.
3. River map, **[name]_r.bmp**, defining where rivers start, end, split and merge.
4. Province map, **[name]_p.bmp**, defining the shape and sizes of provinces.

When creating a tile of a certain size it can be recommended to make a contour. The contour should show the coastline of the tile and should be seen as a rough sketch. This can be done with a 3 pixel wide pencil on a transparent layer.

After the first sketch is done lower the opacity of the layer and draw a new contour on a layer below, this will be the coastline of the tile. Draw this contour with a 1 pixel pencil on white background as its going to be a work in progress and its easier to make changes this way.

When the contour is finally done fill the landmasses with black. The reason for this is that it gives a better expression of the final tile than if you only have the contour. Go over the contour coastline again if its needed.

As the contour is going to be the base of the tile its a good idea to put some work into it.

Now its time to look at the actual parts of the tile.

The text file

Contain the script that tells the game how to read the three other maps. The contour isnt going to be used here besides the fact that we have determined the size of the tile.

- [Name].txt

Look in the existing files to find a tile similar to the one being made. Use that file as a base and make the edits needed so it fits the needs of your tile.

In the following the script in the text file will be explained. It is split into different parts depending on what they affect.

The first part define the overall use of provinces.

Script	Explanation
sea_province = { R G B }	Define [province RGB] as a sea province
lake_province = { R G B }	Define [province RGB] as a lake province
wasteland_province = { R G B }	Define [province RGB] as a wasteland
strait = { from = { R G B } to = { R G B } through = { R G B } }	Define a strait going from [land province 1 RGB] to [land province 2 RGB] through [sea province RGB]
Regions = [number] region = { R G B }	Define number of regions and [province RGB] of one province in each region, the rest will be generated. For the best results if theres more than one region define provinces as far from each other as possible.

Sea provinces should in general be specified along the whole coastline. In larger tiles there should be regions defined. If a tile does not have a region it will join a region from an adjacent tile. Regions also work as the area for the tradenodes.

The next part is the different flavor you can add to provinces. Its possible to add any of the modifiers found in *[EUIVfolder]/common/event_modifiers/00_event_modifiers.txt*, the following are the most used and should be considered for most tiles.

Script	Explanation
river_estuary_modifier = { R G B }	Adds modifier to [land province RGB]
important_natural_harbor = { R G B }	Adds modifier to [land province RGB]
inland_center_of_trade_modifier = { R G B }	Adds modifier to [land province RGB]
province_names = { "[province name]" = { R G B } }	Define a name for [province RGB], which it will always get]

Modifiers should in general not be placed in the same province. Estuaries works well for deltas. The two others are a matter of feeling and will be prioritized for tradecenter placements. Province names should atleast be added to large wastelands or special provinces etc.

After the province flavor has been added its time to look at the factors that can determine how the tile is shown in the game.

Script	Explanation
do_not_rotate = yes	The tile will not be rotated. The tile can still be mirrored.
do_not_rotate_or_mirror = yes	The tile will not be rotated or mirrored.
restrict_to_north_edge = yes	The tile will be place along the northern edge of the map, need for tiles which isnt surrounded by water.
restrict_to_south_edge = yes	As above just placed along the southern edge of the map.
Continent = yes	Defines the tile as a continent. This should be done to large tiles, above 5 x 5. During generation the game will push these tiles north or south to try and fit two.
Fantasy = yes	Define the tile as a fantasy tile. Fantasy "mode" is enabled by standard in the game but can be turned off in options.
Unique = [number]	A tile is per standard defined as unique. Setting this to -1 makes it possible for the tile to appear more than once. Setting this to the same for two tiles will make certain that only one of the two will appear on the map at any given time.

Non of the above is actually needed for the tile to function in most cases. However in some cases a tile wont function if its rotated. An example would be a tile 2304 pixels wide, which is the full width of the RNW. If such a tile is rotated it will exceed the height of the RNW thus stop working. Restricting a tile to the north or south edge is also needed if the tile has a landborder.

At last we have the part of the text file which is needed for the tile to function.

Script	Explanation
num_sea_provinces = [number]	Number of provinces defined as sea provinces. Only sea not lake provinces.
num_land_provinces = [number]	Number of land provinces. Only land not wasteland provinces.
size = { [grid width] [grid height] }	Size in grid squares. So a 256 x 256 is a size = { 2 2 }
weight = [number]	The weight determine how often a tile will appear in the RNW, where 100 is the standard.
empty = { R G B }	The color in RGB of the borderline area in the province map. This area is used for ramdomized seazones.

Ofcourse in the case that theres no land provinces in the tile that can be left out. Finding the amount of land provinces in a large tile can be done through "*Colors → Info → Colorcube Analysis*" which will show the number of unique colors in the file. Remember to subtract sea, lake and wasteland provinces aswell as the "empty" color.

The text file is a work in progress while the three .bmp files are being made. Now its time to take a look at how to actually make a tile. First up will be the height map.

The Height map

Defines the heights of the different parts of the tile, it also defines which parts of the map are above and below water.

- [Name]_h.bmp
- Grayscale – it will be 8 bit
- Using GIMP 2.8.14 requires "Do not write colourspace information" checked when saving.

The height map need to be in Grayscale. This is done via "*Image → Mode → Grayscale*".

An explanation of the Grayscale values can be seen below:

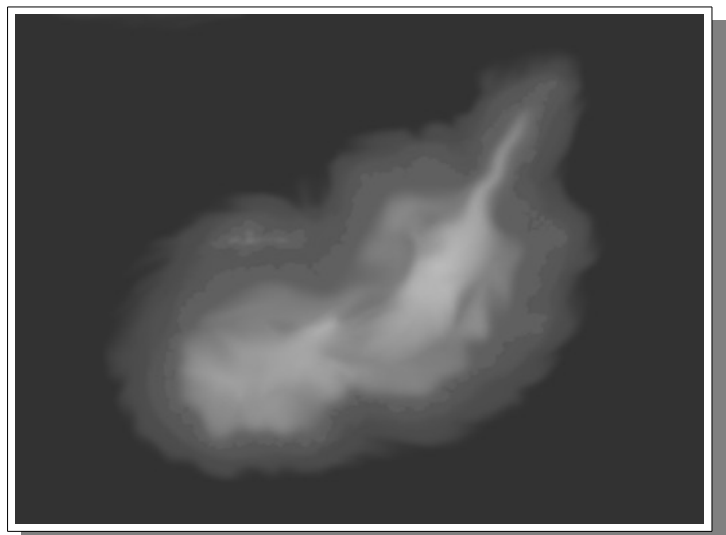
RGB (Grayscale)	Height
50	Deep ocean.
75-88	Gradual fading out into deep ocean.
90-93	Visible shallow water.
94	Highest value below water. Use sparringly.
95	Lowest point above water.
96-180	Most used range.
230's	Highest point on existing map.

The height around value 88 can also be seen as shallow water but is more faded than the higher values.

Transitions from one value to the next needs to be gradual and not too "steep" otherwise the texture will look weird. This is something which is learned with practice.

Wherever a river runs into a lake/sea the height need to get below a value of 90 within the first couple of pixels otherwise it will be possible to see a graphical river "end".

Consider not using value 94 and 95 as



they can lead to glitches. The same is true if landmasses are only 1-2 pixels apart, which can glitch when at a certain zoom level.

Its advisable to use more layers when working on especially the height map. The following is an example of which layers to use, they are shown from top → bottom.

1. One pixel border on coast, height 96
2. One pixel border on coast, height 93
3. Mountain range, height 96+
4. Tile above water, height 96
5. Tile below water fade, 75-93
6. Tile below water, height 50

The borders can be created when having either landmasses or sea selected by "*Select → Border*" then choose a one pixel border. If done for sea be aware that the border has been created around the whole tile and not only along the coastline. This need to be corrected otherwise the edge will be visible on the final tile.

The Nudge tool is the best asset in making a good height map, the blur tool can also come in handy.

The height map can be made a bit rough at first. The important part is getting a feel for where the mountains, valleys etc. are so this can be used for the river and province map. After the tile has been observed in the game changes can then be made, hills leveled out, the height of mountains increased etc.

After the height map its now time to take a look at the rivers.

The River map

Is where you designate the placement of rivers.

- [Name]_r.bmp
- Indexed – it will be 8 bit
- Using GIMP 2.8.14 requires "Do not write colourspace information" checked when saving.

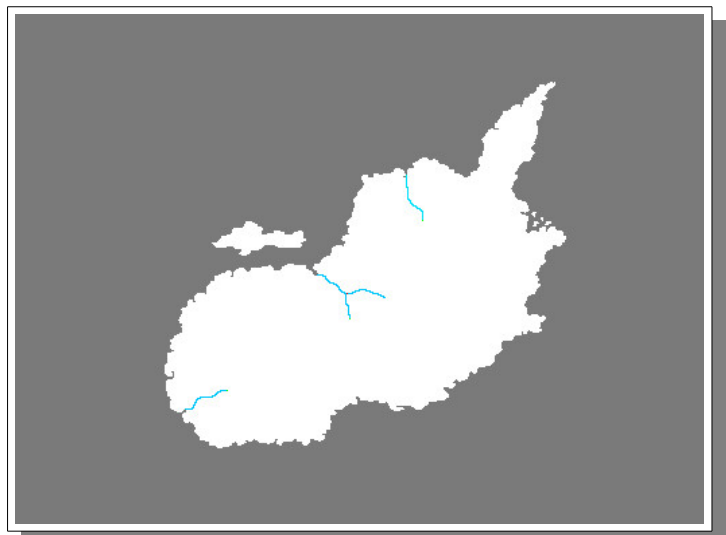
The easiest way to make a river map is copying an existing file. It ensures the file is indexed and gives access to the correct color to create the map.

This can be done through a couple of steps:

1. Find an existing river map. Resize it to the tile needed: "*Image → Scale Image*"
2. Make a layer for the tile
3. Import contour (white landmass, grey sea)

The river map does not need to show landmasses however it can be seen as a help.

If made from scratch indexed is turned on through "*Image → Mode → Indexed*".



Make sure theres enough colors in the colormap for it to be 8 bit.

The colors in the index can be found through: "*Windows → Dockable Dialogs → Colormap*". Use these colors to make the rivers, their function can be seen below:

Color	Function
Green	Beginning of a river. There can be only one for any giving river no matter the amount of merges or splits.
Red	Merge of river. This is used when a side river connects to a main river.
Yellow	Split of river. This is used when a side river splits off from a main river.
Blue (different shades)	Different shades of blue determines different sizes of a river.

A river needs to be one pixel thick all the way. It will cut itself short if it at any point is two pixels or if the pixels only connect diagonally.

Rivers should be drawn into the ocean/lake in which they end, so they "disappear"

After the river map is done its time to look at the province map.

The province map

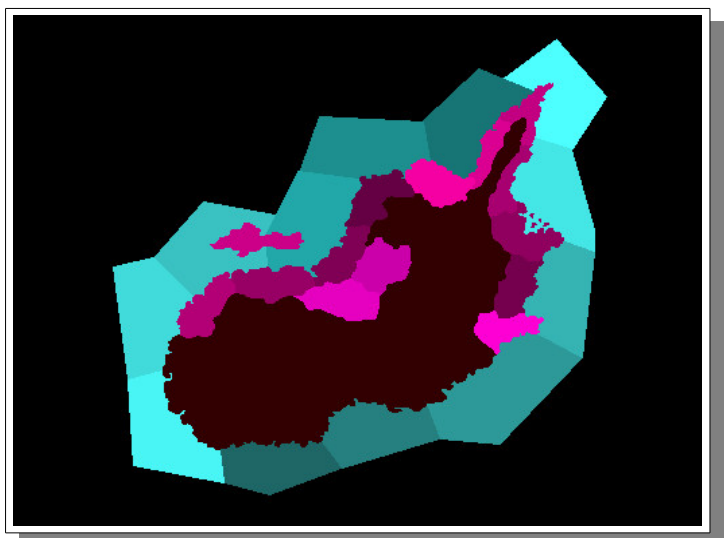
Is the place where the border between the different provinces are drawn.

- [Name]_p.bmp
- RGB – needs to be 24 bit

For larger maps its important to check that GIMP does not save it as a 32 bit file. The tile wont work if this is the case.

Every province has its own RGB color, if the same color is used twice it will be considered the same province even if at different ends of the map.

For smaller tiles picking a color at random is plenty for the larger tiles consider using some kind of system. An example could be starting at Hue 360, Satuation 100, Value 100 then working down from 100 → 40 in steps of 10. Then go to Value 95 working down to 45 in steps of 10. When "done" lower Hue by 10 and restart. Reserve Hue 180 for sea provinces, here you will need to lower the Satuation by 10 after the first go through.



The important part is that its possible to distinguish province borders.

For provinces sizes generally do them bigger than imagined. A good guide for seaprovinces is around 125 pixels long. Smaller provinces are also viable but in general keep the sea provinces large.

Land provinces should be atleast 30 x 30 pixels, which is a small province in Europe. GIMP has a grid function "*View → Show → Grid*" which places a grid of squares 10 x 10 pixels across the layer. Having atleast 8-16 full squares inside a landprovince is a good way to keep them from becoming too small.

It is also possible to look at the average number of land provinces in the tile. An average of 6 – 8 per 128 x 128 pixel is a good rule of thumb, so a 256 x 256 pixel tile has around 24-32 provinces. Keep in mind that the actual tile has a lot to say in regards to the number of provinces.

Now that the files for the tile are finished it time to check it in the game.

Finishing Touch

When adding the tile to the game consider deleting every other tile. You need to delete all files for each tile. So both the .txt and the .bmp files. However DO NOT remove the Water1, Water2, Water3 files. These are needed to generate the sea provinces. This way only the tile thats being worked on will appear in the game.

If the game crashes when generating the new world/launching the new world theres a problem with file types. Either they arent the correct bit, correct format or something along those lines. This needs to be fixed first. Another cause could be a region defined to a color not existing in the tile. Generally check the error file when testing out a new tile. It can be found at ***/Documents/Paradox Interactive/Europa Universalis IV/logs/error.txt***, it will log if theres a mismatch between the defined number of provinces and the actual or similar stuff. So it might just provide the information needed to fix that small mistake.

When the tile works in the game its time to check it for different stuff. If a coast is squarish it is because of the transision is too steep. The same is the case if the texture looks weird on mountains etc.

Go over all rivers to check if some of them come off a bit short, maybe the river is running up hill so it should be redrawn, be shorter etc.

Check land/wastelands provinces to see if the border can be improved. Especially the border between these two types.

Check province flavours/regions do they look good are the tradenodes are they placed as expected.

And that is how a tile is made. Hopefully this guide has been useful, good luck and happy tile making.