

RandomParadox Documentation

P

July 2024

1 Introduction

A tool to randomly generate scenarios for multiple games.

Currently implemented modules:

- Hearts of iron IV (Scenario and MapTool (Generation, Loading, Cutting, Scaling))
- Europa Universalis IV !!CURRENT MODS CRASH ON STARTUP!! (MapTool (Generation, Loading, Cutting, Scaling))
- Victoria 3 !!CURRENT MODS MAY CRASH ON STARTUP!! (Partial Maptool, experimental)

1.1 General Functionality

- Fully random maps with fully random continents, climate, provinces, regions, countries and more
- Generating random terrain and climate from simple input continents
- Generating from detailed input terrain, such as pre-drawn deserts, mountainous areas etc
- Provinces/Region generation, either fully random, or respecting given borders (such as rivers/states/countries)
- Generation of mod files for multiple games, creating loadable mods
- cutting and scaling of regions from the base game
- So much more...

1.2 Some general reminders

- You need to drag in images of the correct resolution
- **If the GUI isn't responding, it is probably simply busy.** Some calculations need time, check the console window that opens with the tool to see live updates.
- **Don't click the GUI while it is busy!**
- Keep in mind that you can almost always first have something generated, then modify it with your favourite editor, and then drag it in again. This can make your life a lot easier if you want to prepare certain inputs, such as climate, provinces, states or countries (and more)
- It can happen that the tool crashes
- If the tool no longer works like expected, you can click the **reload config** button in the first tab to reset it to the initial configuration

To avoid error reports that are due to incorrect configuration, please read the documentation carefully.

The GUI will not respond in cases where it has to calculate stuff. This might sometimes take minutes for some difficult calculations such as climate. Please just wait and keep the separate console window in mind, this one updates constantly.

2 GUI

2.1 Overview

The GUI consists of a log window, generic settings and buttons, and tabs. The start window will look like this:

At the top, you can see the log messages, keep an eye on them in case of errors. They will only update after calculations are done, so check the console window that opens with the GUI for live updates.

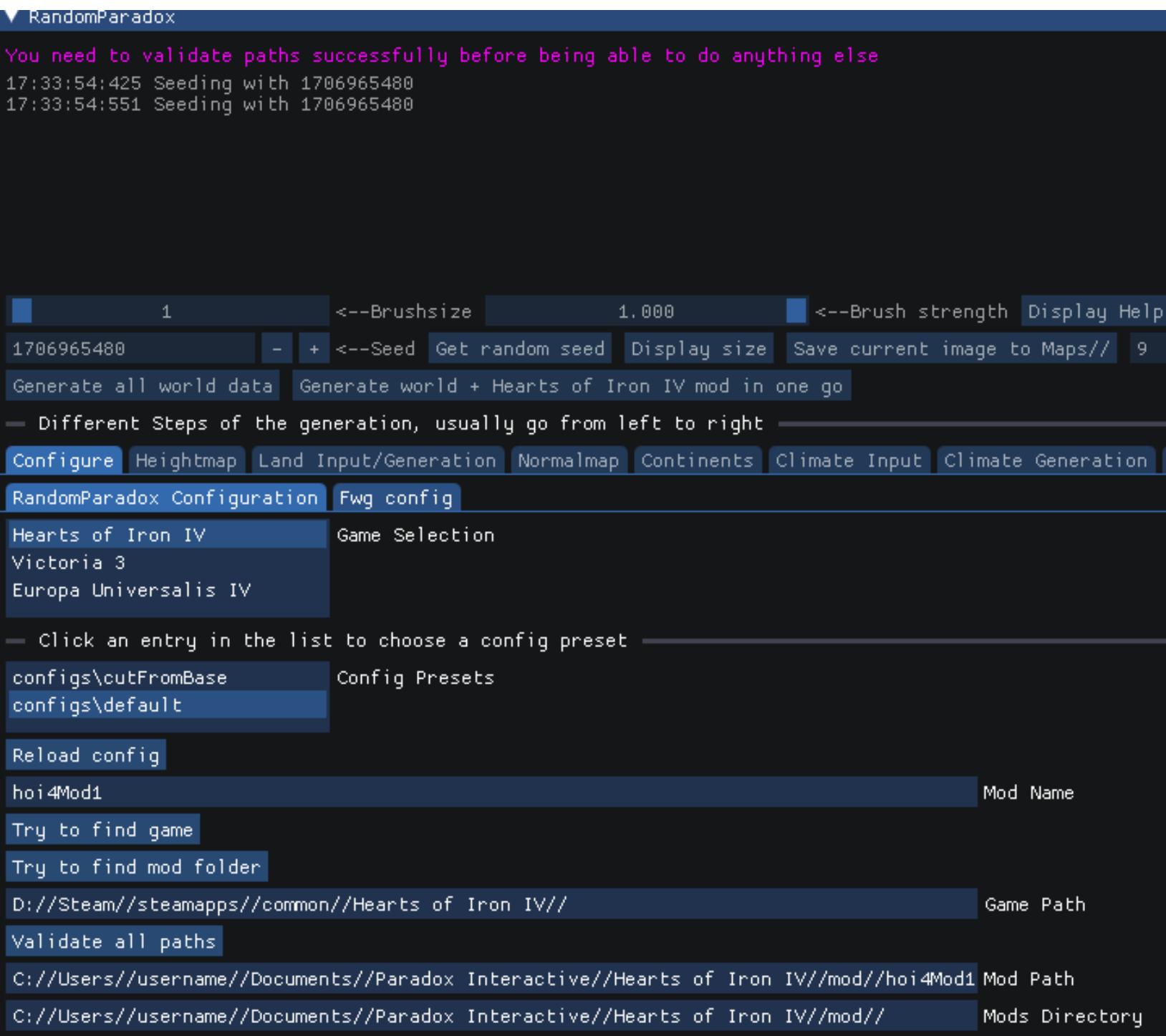


Figure 1: Overview tab

2.2 Configure

In the configure tab, you can set multiple values, and have some automatically set, if possible. First of all, you can select the game you want to generate for, and select a config preset from the list.

There are two config presets:

- **default**: This preset is meant for the mixture of generating and loading maps, you basically go through all the tabs one by one and generate and load, and configure all the desired settings in one tab
- **cutFromBase**: This is meant to cut an area from the base game files. This works for Hearts of Iron IV. Go into the heightmap tab, there you find coordinates you can set, and define an area of the map you want to cut out, and potentially scale to another resolution. For a detailed explanation, see chapter 3.

You can try to find the game via the **Try to find game** button, and try to find the mod via the **Try to find mod folder** button. You can also set the mod name.

Before you can proceed, you will have to successfully validate paths by clicking **Validate all paths**. If it is successful, the other tabs will become clickable.

2.3 Heightmap

Here, you can generate a heightmap or load one by dragging in an image. You also can skip this step if you have a terrain map you want to input, you can do this in the **Land Input/Generation** tab.

When generating an image, you have a large amount of settings available, and can also manipulate settings for individual layers of the heightmap.

2.3.1 Individual layer settings

Heightmaps are generated from multiple combined layers. Each layer has different properties:

- **type**: the fractal type that is used
- **landLayer**: if this layer will be applied only to land after the basic continent shape has already been determined
- **fractalFrequency**: the higher, the smaller the fractals
- **fractalOctaves**: affects ruggedness
- **fractalGain**: how rugged the generation is

- **seed:** individual seed for a layer, if 0, it is automatically determined based on the base seed
- **widthEdge:** from when on terrain height falls towards the east/west edges of the map
- **heightEdge:** from when on terrain height falls towards the north/-south edges of the map
- **weight:** how much this single layer is weighted relative to the other layers
- **maxHeight:** what the total maximum height present in this layer can be. Should be between 1 and 255
- **minHeight:** what the total maximum height present in this layer can be. Should be between 1 and 255, and smaller maxHeight
- **tanFactor:** how quickly land height increases. Can produce really rugged layers, specifically good for e.g. mountain ranges. Range between 0.00 and 0.26

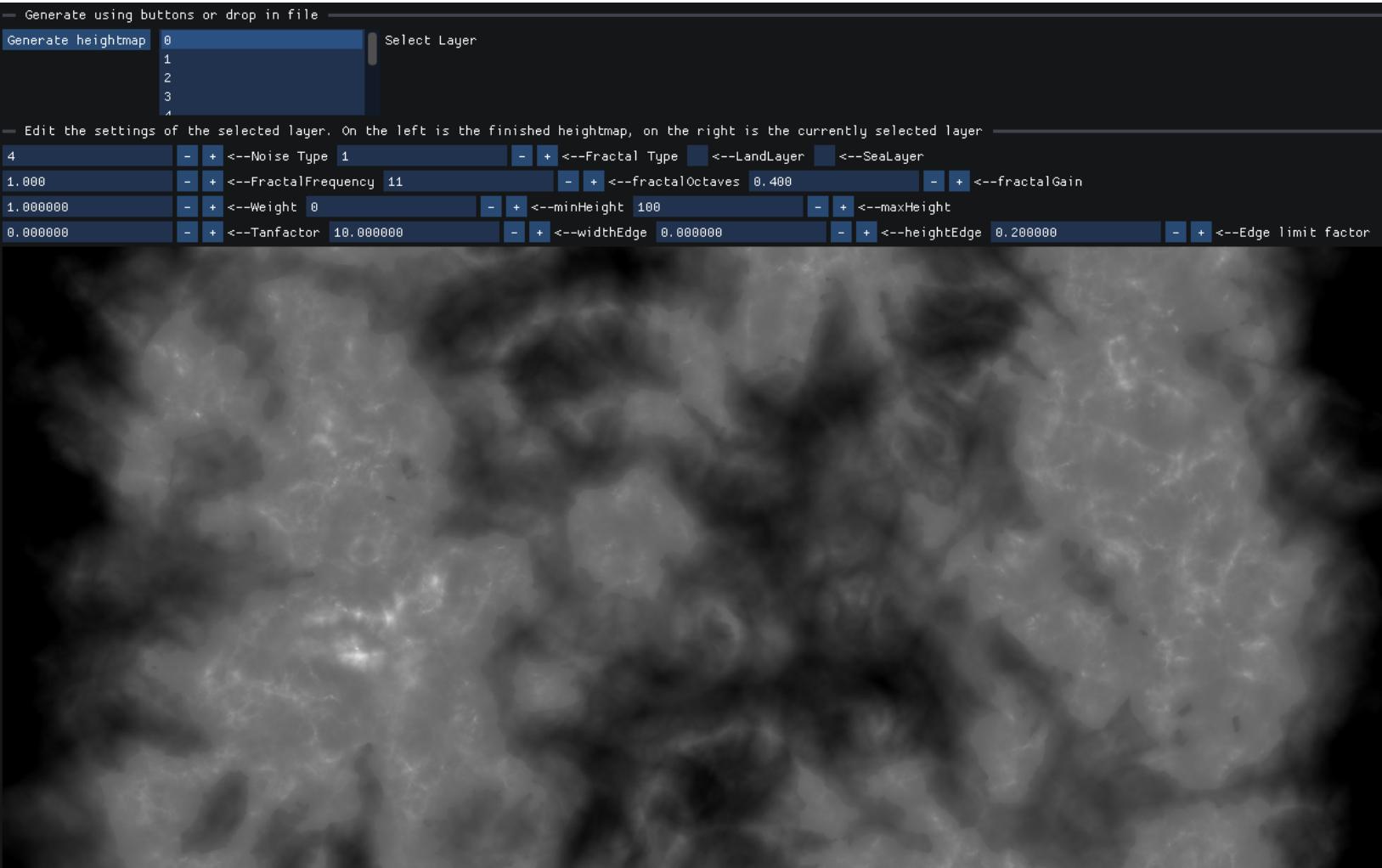


Figure 2: Heightmap Generation or Input

2.4 Land Input/Generation

This tab has multiple functionalities:

2.4.1 Land Generation from a Heightmap

This simple case is done when you simply want to derive a landmass from the heightmap. For this, simply set **seaLevel**, which influences the overall shade of the map and **landPercentage**, which decides how much land will

exist. Then simply click generate and observe the result.

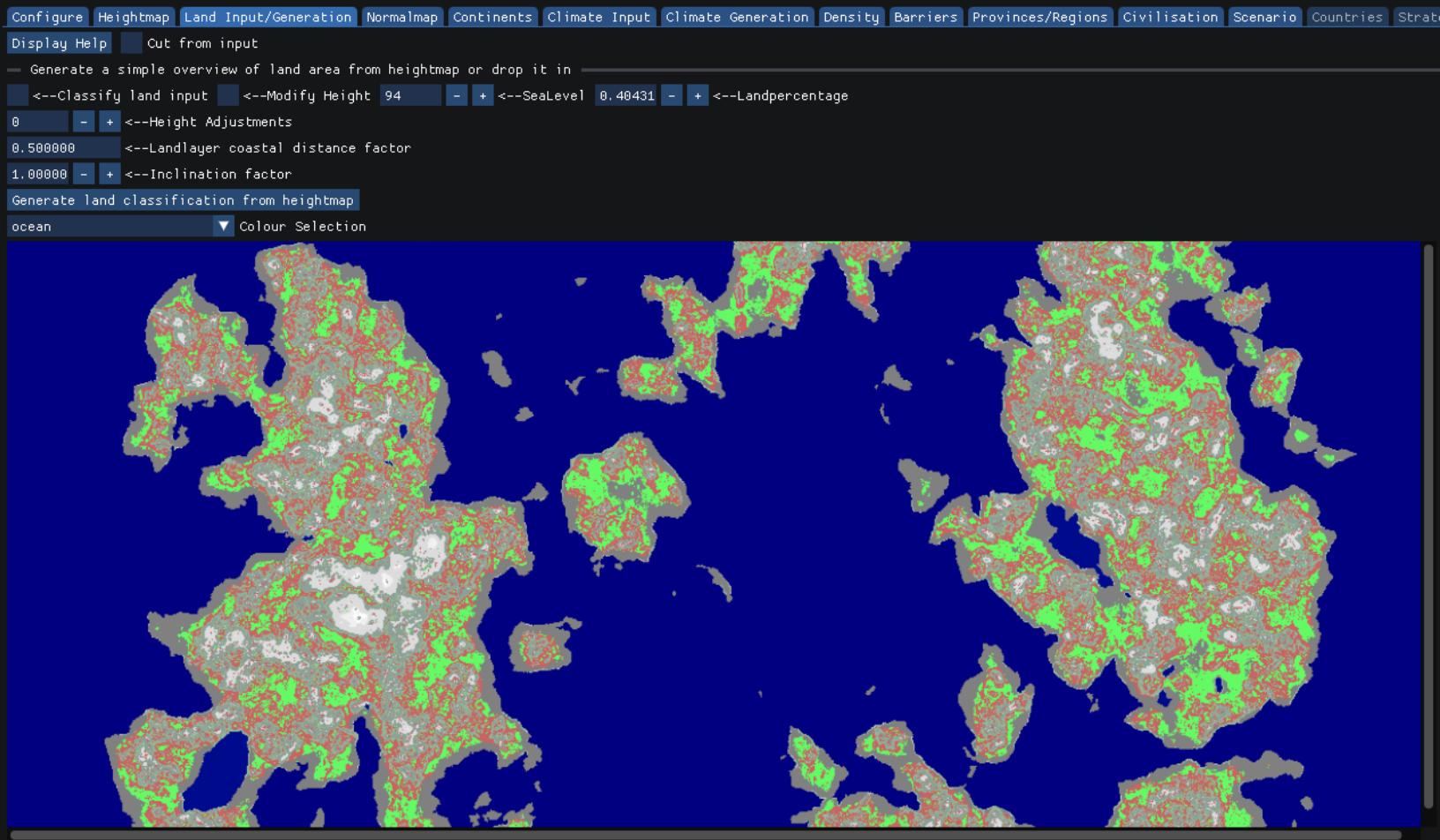


Figure 3: A land shape, generated from a heightmap

2.4.2 Heightmap Generation from Simple Input

Instead of generation a land mass from a heightmap, you may also derive a heightmap from a given land mass. For this, drag in an image. The tool will then try to derive automatically what is land and what is ocean. If in doubt, try to stick to the colours which are otherwise generated. See figure 4 for an example

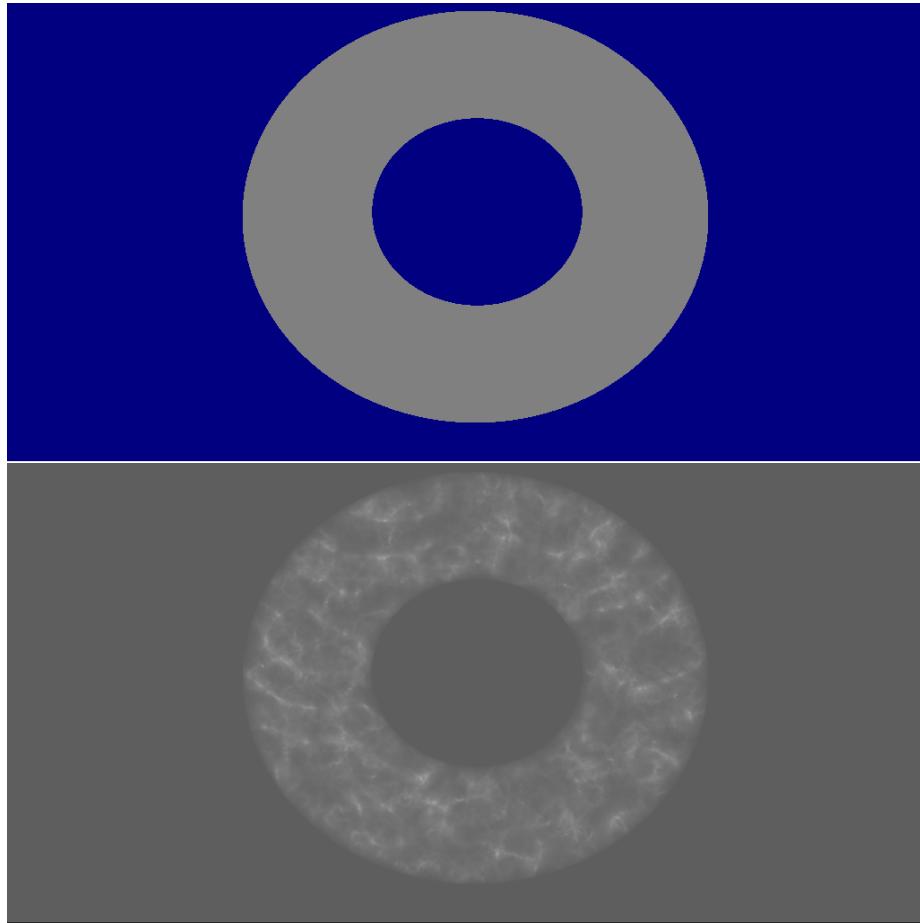


Figure 4: The dragged in input + the resulting heightmap from this input

2.4.3 Heightmap Generation from Complex Input

This extends the Simple Input case. By checking the **Complex Input** box, you enable this feature. Now, when dragging in an image, you can afterwards analyze an image and label every single colour to a land type.

Make sure you have a good image without washed up borders, e.g. don't draw an image with a brush, but use a pencil. You might otherwise get far too many colours to label, this would be tedious.

See figure 13 and 6 for an example.

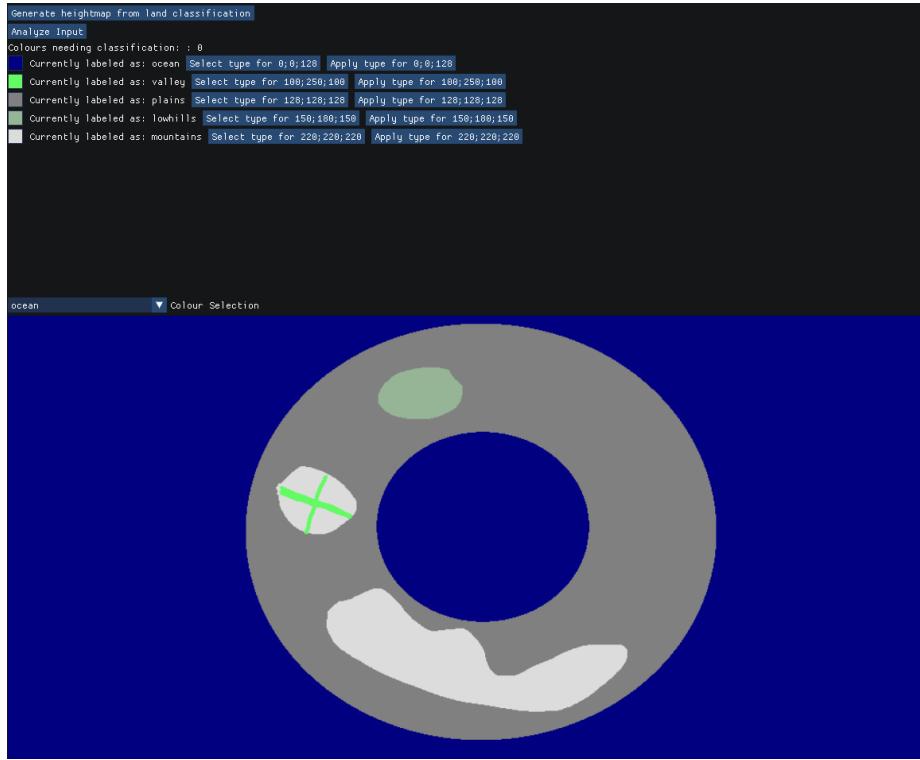


Figure 5: The dragged in input while labeling the colours.

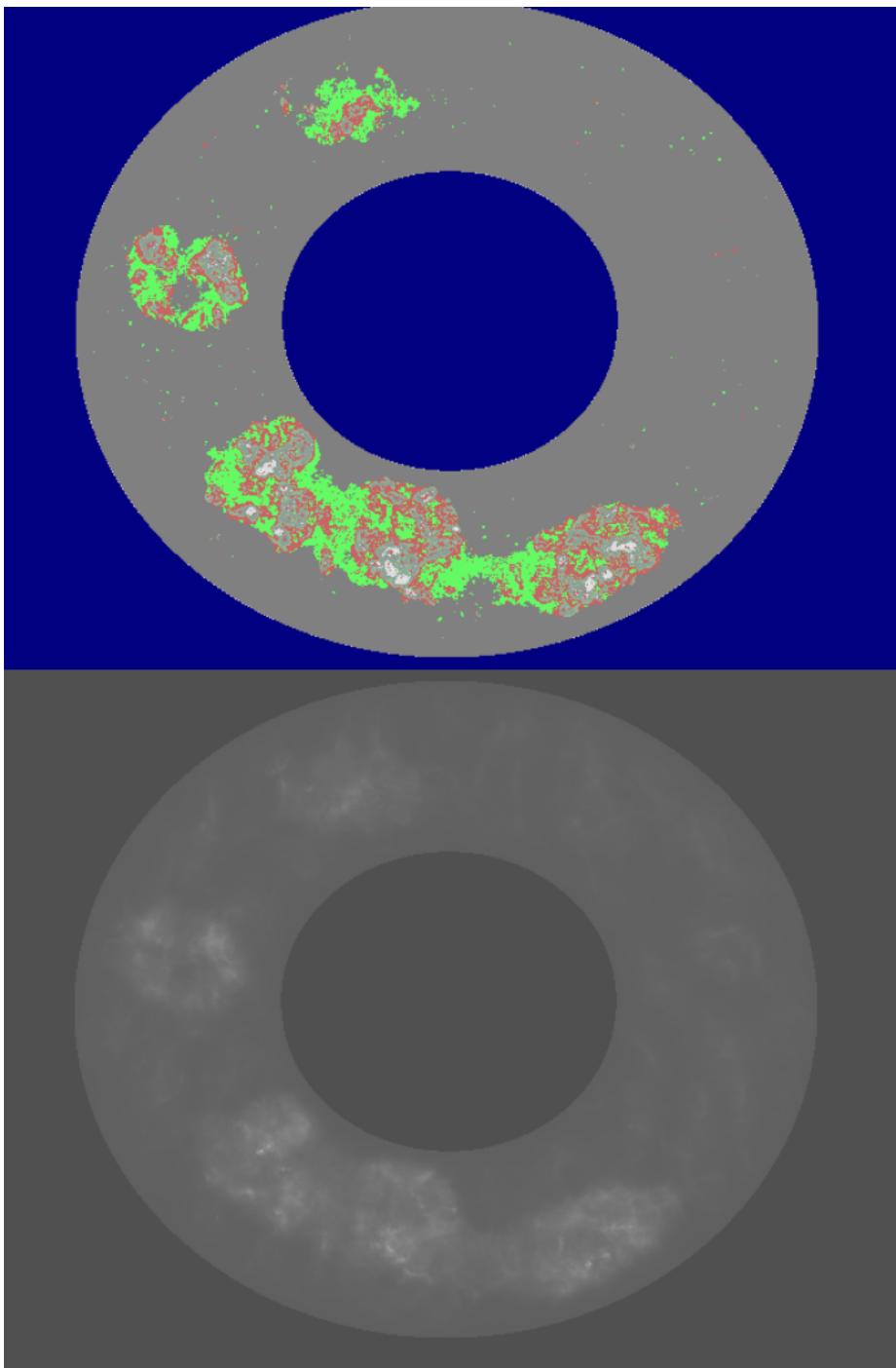


Figure 6: The resulting heightmap after labelling. Here you can clearly see the areas which were designated as mountains, hills, valley, or plains.

2.5 Normalmap

You can generate a normal map in this tab. Modifying the **SobelFactor** makes the land bumpier or smoother. For an example see figure 7.

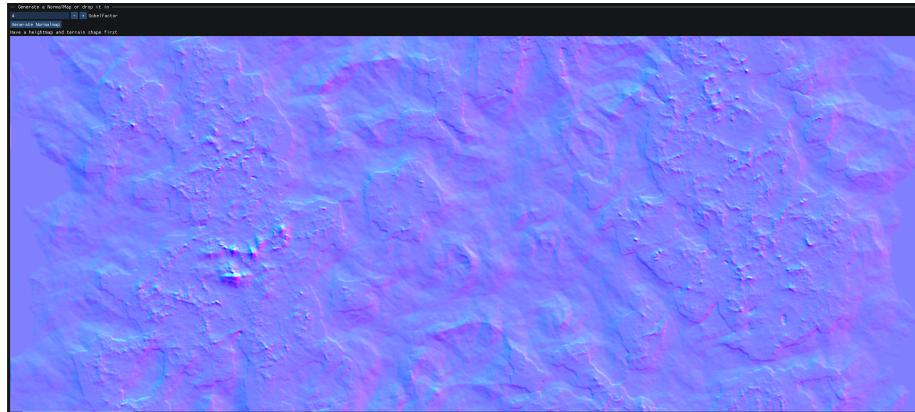


Figure 7: Generated Normal Map

2.6 Continents

This shows an overview of the continents that were detected. You can also input a map of continents yourself, black $\text{rgb}(0,0,0)$ is water.. This influences provinces and states. For an example see figure 8.

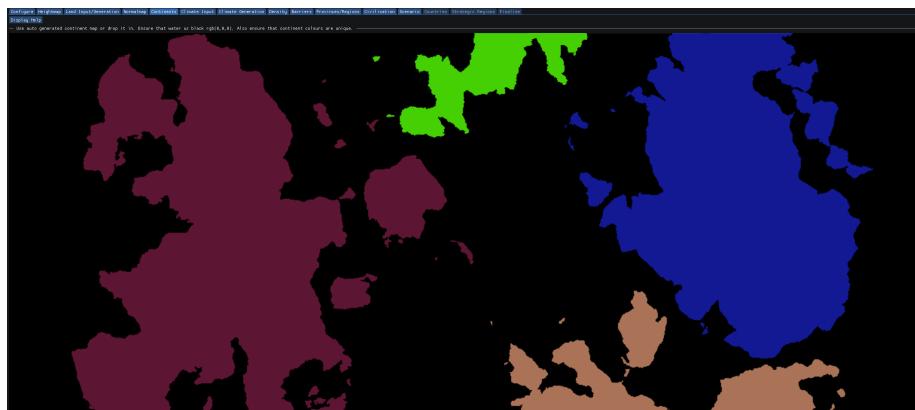


Figure 8: Generated Continent Map

2.7 Climate Input

An optional tab, in case you want to specify climate types instead of generating them. To do this, drag in an image of climate zones, and label them accordingly with the given climate types that match best. The tool will then derive the rest from these zones, such as province density, see 9.



Figure 9: This shows the tab with climate input, and the labelling functionality.

2.8 Climate Generation

This Tab has multiple sub tabs. However, you can automatically generate everything by clicking **Generate Climate Automatically**. The factors to modify are:

- **Base humidity:** How humid the overall world is. The lower, the drier.
- **Base temperature:** How hot the overall world is. The lower, the more ice. The higher, the more jungle/desert
- **Fantasy Climate Frequency Modifier:** The higher this is, the smaller each random area is
- **Fantasy Climate:** This creates completely random climate, which follows no real world rules
- **River effect range multiplier:** How wide the effect of rivers on humidity and therefore climate is
- **River humidity multiplier:** How strong the effect of rivers on humidity and therefore climate is
- **River amount multiplier:** Affects how many rivers there are
- **Latitude high:** The upper bound of the latitude. $2.0=90$ degrees north (North pole). Equator at 1.0
- **Latitude low:** The lower bound of the latitude. $0.0=90$ degrees south (south pole). Equator at 1.0

For an example, see figure 14.

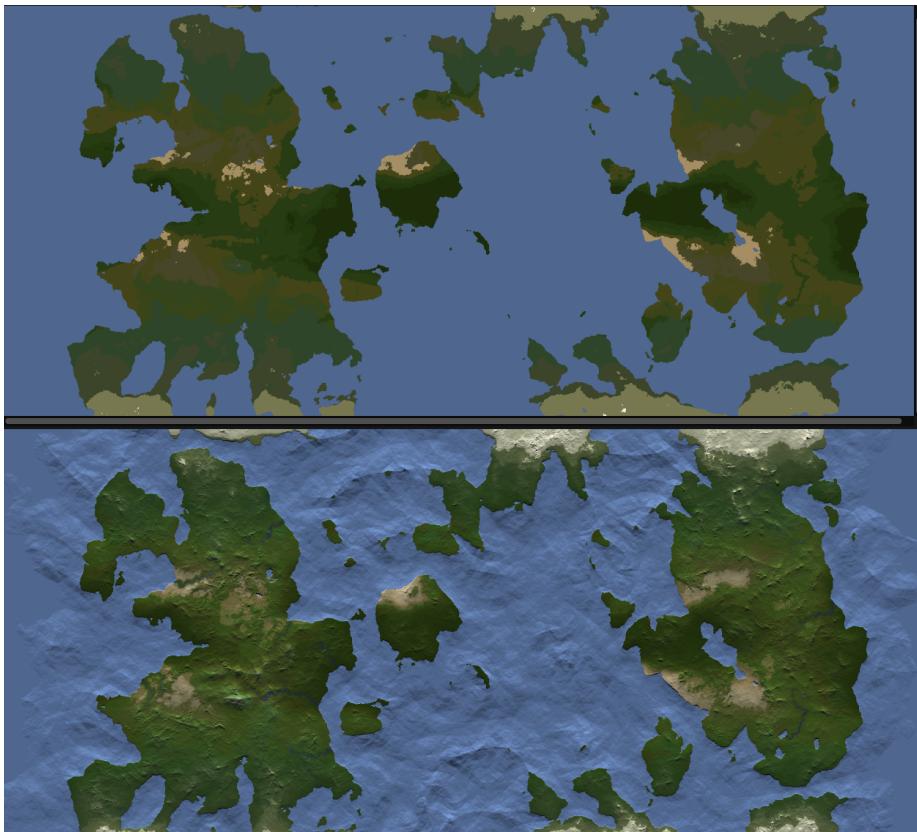


Figure 10: Generated Climate Map

2.8.1 Temperature tab

Temperature is a major factor for climate generation. If you don't want to handle climate zones in detail, you can simply drop in a temperate map and generate climate classification based on it.

2.8.2 Humidity tab

Humidity is a major factor for climate generation. If you don't want to handle climate zones in detail, you can simply drop in a humidity map and generate climate classification based on it.

2.8.3 Rivers

You can modify a factor on how many rivers you want. Rivers also affect climate generation, see the factors listed above in the climate overview.

2.8.4 Climate

You can either generate a climate map or drop it in. When dropping in a climate map, you need to make sure it uses compatible colours, that the generator uses itself. Therefore, it might often be best to first have it generated automatically, then modify it, then load it again.

However, you can also use the Land Input/Generation tab to input a map that has incompatible colours and label them there. Go back to section 2.4.3 for an explanation.

2.8.5 Trees

Here you can create a tree map. This will later be used by the games to determine graphical tree coverage. You can increase density of trees in different zones with the density factors listed here.

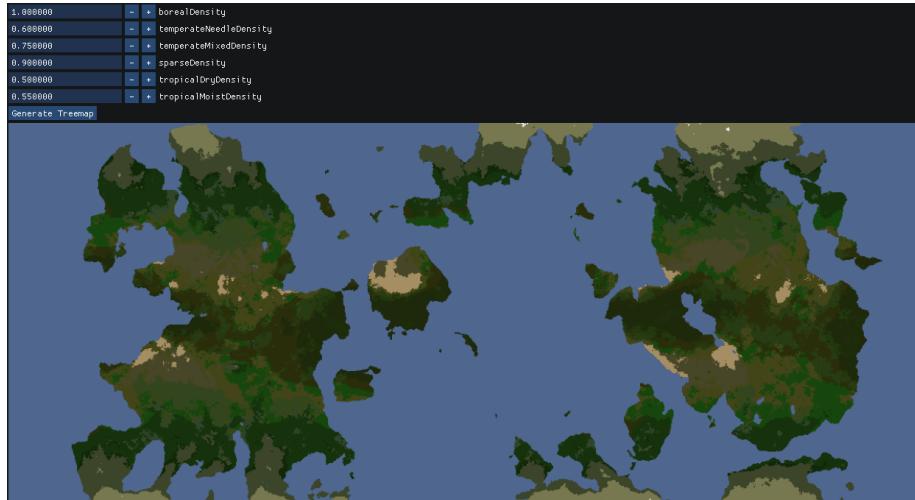


Figure 11: Visualisation of Tree Coverage on climate map

2.9 Density

This tab shows the density for provinces, which is derived from the climate zones. You can also drag in this image to set density yourself.

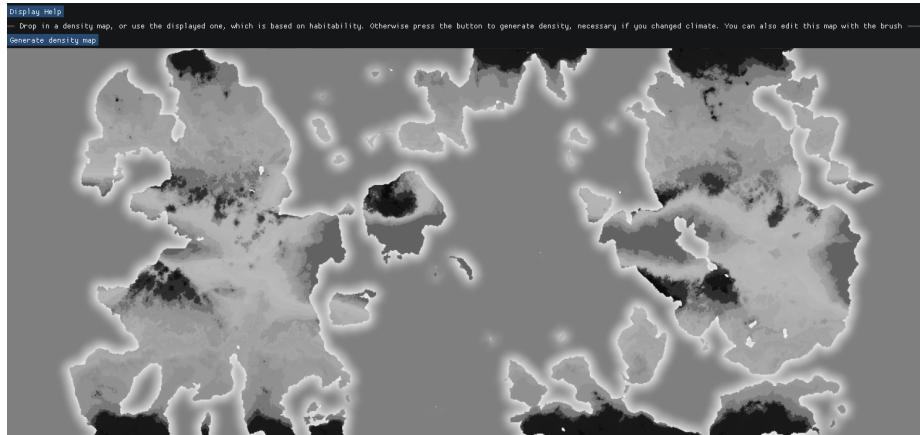


Figure 12: Generated Density Map

2.10 Barriers

This tab allows input of maps that affect later province/region generation. To do this, you have to drag in an image, on which an edge detector will be run. You can select if this is region input or country barrier input. It is best to drag in countries with clear colour borders, if you prepare them, best use a pencil and not a brush.

Region Barrier Input affects generation of provinces, as provinces will respect these given borders.

Country Barrier Input affects generation of provinces and regions. Provinces will directly respect these borders, while region generation will also try to respect these borders. **Please note that these inputs are only for barriers, they are not a replacement for region/country inputs**, this is done later.

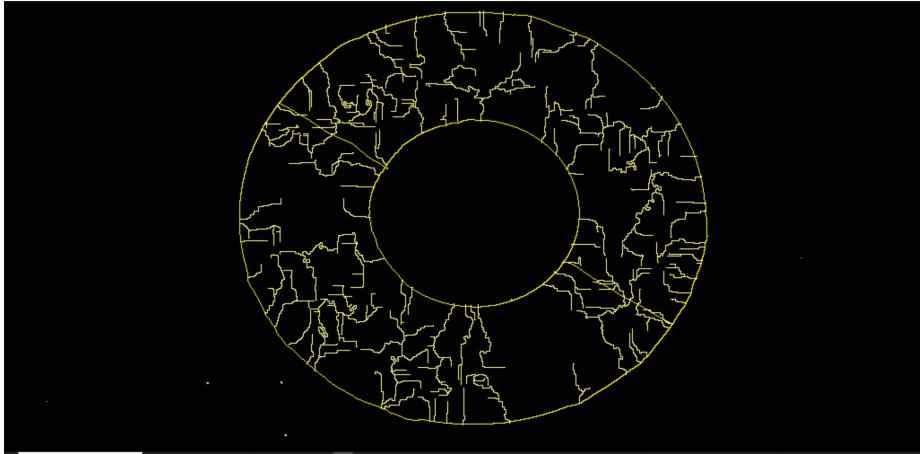


Figure 13: A barrier map resulting from rivers and a dragged in country map.

2.11 Provinces/Regions

This tab shows two subtabs, for province and for region generation.

2.11.1 Provinces

Here you can configure province generation and generate provinces until you are happy with the result, or load a province map. Most important factors are:

- **Landprovincefactor:** The higher, the more provinces on land
- **Seaprovincefactor:** The higher, the more provinces on oceans
- **Density Effects:** The higher, the more the density map influences density and size of provinces
- **Minimum size of provinces:** The higher, the more pixels a province will have
- **Maximum amount of provinces:** An upper limit to province amount, increasing this drastically can have major effects on the generated game modules. **For Hearts of Iron IV, this upper limit should not be larger than 20000, otherwise visual bugs start to appear!**

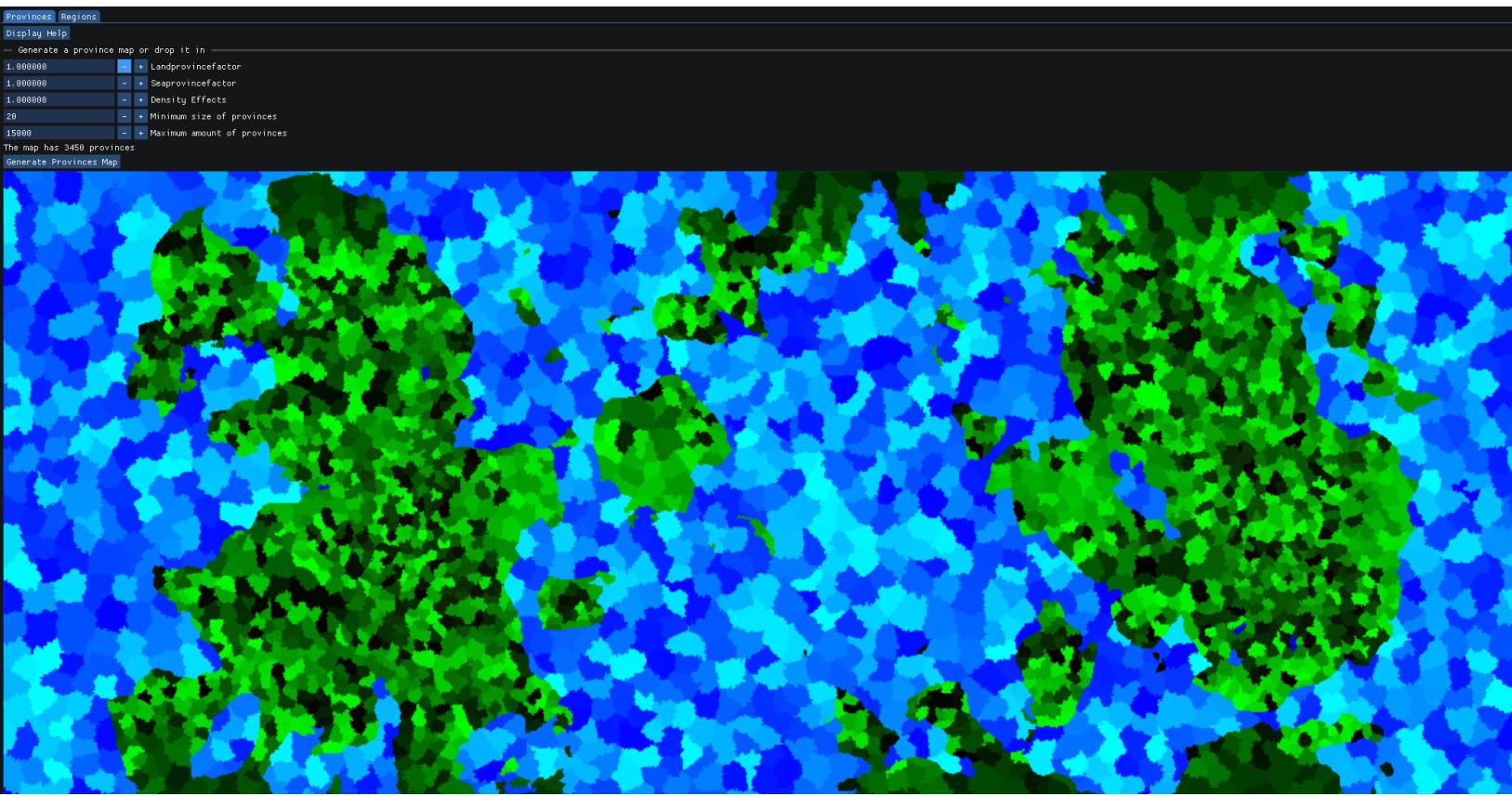


Figure 14: Generated Provinces Map

2.11.2 Regions

Here you can configure region generation and generate until you are happy with the result, or load a region map. Most important factors are:

- **minProvPerLandRegion:** The higher, the more provinces per land region
- **minProvPerSeaRegion:** The higher, the more provinces per sea region
- **targetLandRegionAmount:** The higher, the more provinces per land region. When updating this value, minProvPerRegion will automatically update.

- **targetSeaRegionAmount:** The higher, the more provinces per sea region. When updating this value, minProvPerSeaRegion will automatically update.

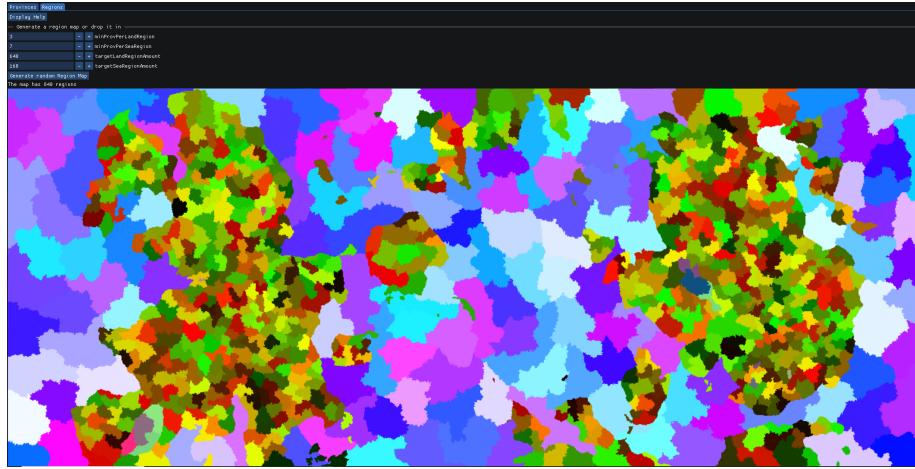


Figure 15: Generated Region Map

2.12 Civilisation

Here you can generate and edit the development, population, city and farm distribution in four sub-tabs:

- **Development:** Affects industrialisation and cities.
- **Population:** Affects population amount and city size. You can draw values for single provinces here, using the brush settings at the top.
- **Cities:** Affects world map look and scenario modules make use of this data. You can draw in this map. The degree of urbanisation can be modified by changing **Pop Influence on city size** and **Urbanisation factor**.
- **Agriculture:** Affects world map look and scenario modules make use of this data. The amount of farms can be changed by the **Agriculture factor**.

2.12.1 Development

While the checkbox **Random Development** is checked, the Button Generate Development will randomly change the base development modifier of

the continents. When turned off, you can manually set the Continent Development modifiers. Development is still modified by climate, so increasing it to 1.0 does not mean completely developed continents, but the factor can be higher than 1.0.

You can draw in this tab, either by setting values for whole continents or by enabling **Draw mode** and setting it for single provinces.

2.13 Scenario

This tab allows setting of options for the selected game module. Dependent on the game, different options show up. You also have to use the **init** button to prepare the rest of the tabs for generation.

2.14 Countries

This tab allows configuration of the number of countries you want generated. You can also select 0. You can also input a list of countries and a country image.

2.14.1 Country Text Input

The list of countries has the format **r;g;b;tag;name;adjective**. If a country has no input mapping, a random country will be generated for it.

2.14.2 Country Image Input

You can input a country image, the states will be assigned as good as possible to match the borders of the countries. Make sure that you have unique colours for countries, so when drawing an image, **don't use a brush, use a pencil!**. You can also first generate a country map, then edit it with your favourite image editor, and then drag it in again. You can also use country input in the barriers tab.

2.14.3 Country editing

All the loaded or generated countries can be edited. See 16 for the buttons that pop up.

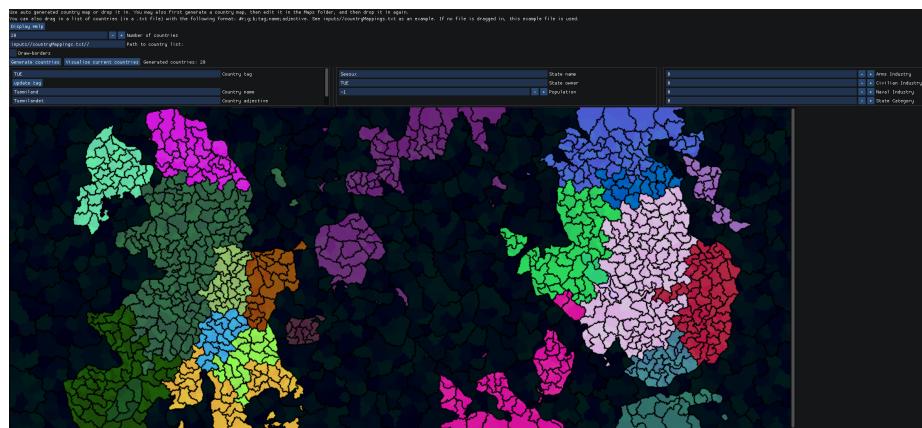


Figure 16: Generated Countries Map

2.15 Strategic regions

You can generate, load or edit strategic regions with the drawing functionality in this tab.

3 Cut and Scale

You may select areas to cut from the base game files. For this, select the **cutFromBase** config preset. Then, drag in the games heightmap into the heightmap tab. You can then see, which area is cut. In the example 17, you can see this in action.

The scale option sets to which resolution all the files should be scaled.

Once satisfied with the area, click the **Generate World + Hearts of Iron IV Mod in one go** button. This will generate some files and cut out others from the game files. A result can be seen in 18.

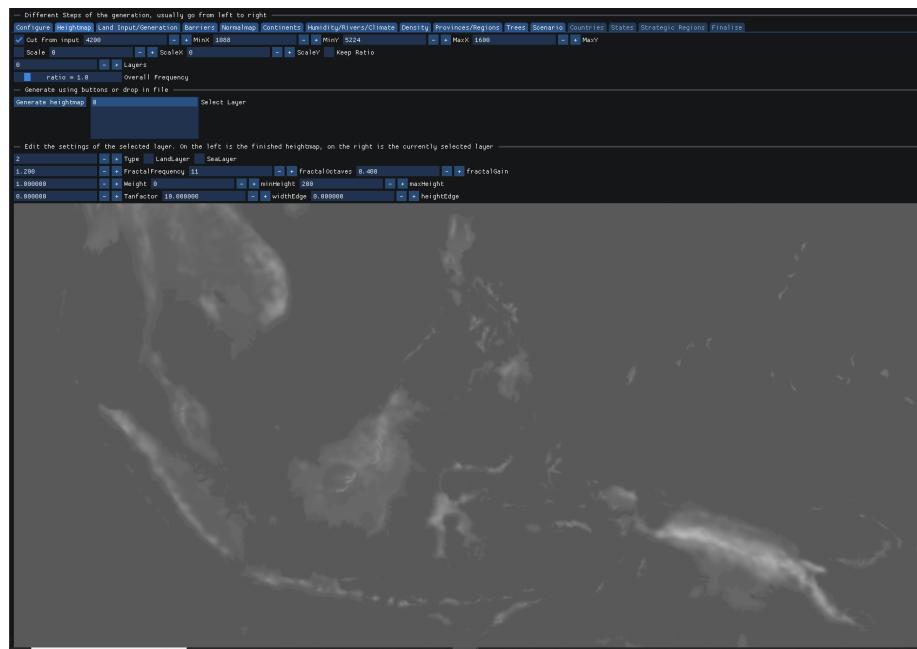


Figure 17: The cut out area of indonesia from the Hoi4 heightmap.



Figure 18: The cut out area of indonesia ingame

4 Modifying Resource Files

Warning: This chapter in general is addressed at users that are familiar with the functionality of the generator and want to help expanding it. You should never randomly modify files here.

4.1 Flag Generation

Experienced Users: Only experienced users or modders should modify files here. Stability of the generated scenario can be affected, and crashes of the generator can be caused.

Flag generation works as follows:

4.1.1 Flag templates

First, a flag preset is selected from the "resources flags\flag_presets" folder. Together with the preset, the sidecar .txt is loaded. Such a file contains lines like:

Listing 1: Flag Template Sidecar File

```
#colourGroups;symbolColourGroups;symbolPlacement;symbolX;symbolY;symbolSize;  
p1,p2;s1;true;0.025;0.35;0.25;
```

This defines the colour groups used for this template, in this case only two colours are used, one randomly chosen from group p1, one from group p2. Next, the colourGroup for the symbol is chosen. Next, the symbol placement is set to true or false. True allows a symbol, false does not allow a symbol for this flag type. Next, the x and y offsets for the symbol and the size of the symbol are defined. Note that the offset points to the lower left corner of the symbol **template**. Symbol templates often have larger dimensions than initially appears.

4.1.2 Symbol templates

If the flag preset allows a symbol, a symbol template is selected from the "resources flags\symbol_presets" folder. Together with the preset, the sidecar .txt is loaded. Such a file contains lines like:

Listing 2: Symbol Template Sidecar File

```
#replaceColour  
true;
```

This first option simply defines if the colour should be replaced. For example, for complex insignia, changing colours would give ugly results. In this case, setting false should be preferred.

4.1.3 Adding and editing flag and symbol templates

Any template can be edited, and new templates can be added. New templates must be tga-files in the 82x52 dimension for flags and 52x52 dimension for symbol templates. The filename must be a number, that follows the other templates in the folder. Don't forget to add the corresponding side-car text files.

4.2 Name Generation

Experienced Users: Only experienced users or modders should modify files here. Stability of the generated scenario can be affected, and crashes of the generator can be caused. When modifying name generation, you have different options:

4.2.1 Add new state names in state_types.txt

It is important to format them correctly. Separate them via ; and also ALWAYS contain template where the country name is supposed to be.

Example: **Revolutionary template;** However, you cannot add new ideologies/lines like democratic or monarchy. So no new lines.

4.2.2 Editing token groups in token_groups.txt

You can add token groups by adding a new line, e.g. :

mytokenGroup;n;tt;r;s;reallyLongEntry;

First, the name of the group. All entries afterwards are characters that appear randomly if you use this group in name_rules.txt. Each entry can be as long as you want.

You can of course also edit all other token groups, remove and add characters, or remove a group altogether. However, if you remove a group, you need to remove it from name_rules.txt

If you ever see an error mentioning a missing namegroup or token group, make sure to check for errors in the text files. There is no need to for capitalization of letters.

4.2.3 Editing name rules

You can modify or add name rules. Name rules are randomly chosen for countries/regions etc. You can add as many name rules as you wish. Name rules work like this: For every entry separated by ; in the name rule, a random entry from a token group is selected. E.g.: E.g., the name rule: vowels;groupMiddle;vowels;groupEnd; gives:

- a random element from vowels, e.g. A.

- a random element from groupMiddle, e.g. lt.
- a random element from vowels again, e.g. e.
- a random element from groupEnd, e.g. stan.

The full name now is Altestan.

4.3 Hoi4 Unit Generation

4.4 Hoi4 Focus Tree Generation

Only for experienced modders: Working on the focus tree generation is only possible if you understand focus modding in general.

Focus tree generation is done dynamically. For this, resource files from *resources/hoi4/ai/national_focus* are loaded. The folder structure is as follows:

- baseFiles contain the key-value pairs for the **available**, **bypass** and **completionRewards** fields in a focus. foci.txt contains information on the foci to be used, and their implemented types in the generation. focusBase is the base template file for every focus tree
- the chains folder contains the chains that are evaluated for every country.
- focusTypes contains the foci templates, defined in foci.txt

4.4.1 Flow of focus generation

Focus generation works with chains. A chain consists of multiple steps, that get evaluated. Each chain step consists of multiple fields:

- chainID: a unique ID for this chain.
- chainStep: Unique ID for the step, used as reference by other steps
- stepRequirements: multiple fields, which are dynamically evaluated.
 - predecessor: names the preceding foci. If the preceding focus could not be filled, this focus won't be created.
 - skippable: Names the preceding foci IDs that can be skipped. Skippable foci are therefore optional in a focus tree.
- rank: Determines which kind of country is allowed to have this focus. Options are **weak**, **regional**, **major**, **any**
- ideology: Determines which ideology the source country must have. Options are **fascism**, **communism**, **democratic**, **neutrality**, **any**

- type: Specifies the type of the focus, must be one defined in foci.txt. This specifies the template file being used from the focusTypes directory
- target: The requirements the target must fulfill:
 - rank: Determines the rank the destination country must have: Options are **weak, regional, major, any**
 - ideology: Determines the ideology the destination country must have, **relative to the source country or the specific ideology**: Options are: **not, same, any, fascism, communism, democratic, neutrality**
 - location: How far away the destination target must be. Options are **neighbour, near, far, any**.
 - target: If this country can be selected multiple times on the same level or chain. If for example stepID 0 gives an annex wargoal against a country, step 7 shouldn't give it against the same country again. Options are **notlevel, level, notchain, chain**
- relation: Fields that specify the relations between foci.
 - Exclusive: specify stepID with which this focus is mutually exclusive
 - and: specify stepID which is also required to be fulfilled. If e.g. focus 5 has *predecessor{3,4}*, and 3 has *and{4}*, 4 has *and{3}*, both 3 and 4 must be fulfilled for focus 5 to be available.
 - if nothing is specified in both fields, multiple predecessors are considered as an **OR** relation, meaning any of the preceding foci is enough to proceed in the focus tree
- Date: The date from which on the focus is available: Format: DD-MM-YYYY, e.g. 01-01-1937.
- available, bypass and completion rewards: Defined here are **keywords**, which specify **values** in the *baseFiles/available.txt*, *baseFiles/bypass.txt* and *baseFiles/completion_reward.txt*. These values are explained in the next section.
- level: the last field specifies on which level the focus is. This means that e.g. if you want a focus that has an exclusive relation, e.g. ally fascist state A or annex fascist state A, both are on the same level, so that *targetlevel* can specify that the target MUST be on the same level, so you can guarantee that the SAME country is targeted twice.

4.4.2 Available, bypass and completion_reward files

These files contain key, value pairs of hoi4 national focus code. They key{} field is used in the chain files, to look up the longer value{} fields. An example is given here:

Listing 3: Flag Template Sidecar File

```
key{coastalState}
value{any_state = {
    is_coastal = yes
    is_controlled_by = ROOT
}};
```

In this example, if a chainstep specifies `available{coastalState}`, this code is placed in the focus. The content here is Hoi4 code, specifying that the country must have at least one coastal state.

This functionality works the same with bypass and completion rewards. Specify the keys in the files together with values, and you're done.

Additional note: Also usable are template keywords: If you for example specify `templateSourceTag` or `templateDestTag`, these will be replaced by the source or destination tags during the final creation steps of the focus tree generation.

4.4.3 Issues and lacking functionality

So far, no relations to other chains are possible, only internally in a chain. So you can't specify that a chain is exclusive with another one. In that case, you'd have to build a larger chain incorporating the smaller chains, to work with the step relations.

Furthermore, you can't specify that a focus E has as predecessors (A and B) OR (C and E).

More complex relations might also not work. If you find a relation that doesn't work, please forward it to me.

5 Common Issues

5.1 Common errors

- “Configured paths seem to be messed up, check Hearts of Iron IV Module.json”: You have misconfigured paths. Make sure they are in a format like `D://Steam// steamapps//common//Hearts of Iron IV`. The double forward slashes “//” are **important**
- Incorrect config “*.json”: The mentioned json file has an error. The line the error is at should also be mentioned in the message your pro-

gram shows. Make sure you follow the rules in section ???. If it doesn't work, search for "json validator" and copy the whole file content there.

- "No input map found under heightmapPath, please correct the path": Find and correct the field *heightmapPath* in RandomParadox.json.
- in case the generator can't load one of the maps you edited, make sure you edited them correctly, with the correct tools. See ?? for details.

6 Known Issues

6.1 Necessary Improvements for FastWorldGen

6.2 Necessary Improvements for Hearts of Iron IVModule

- Focus Tree Generation is lacking features and proper focus trees
- Flag Generation needs more symbol templates, colour combinations and flag templates