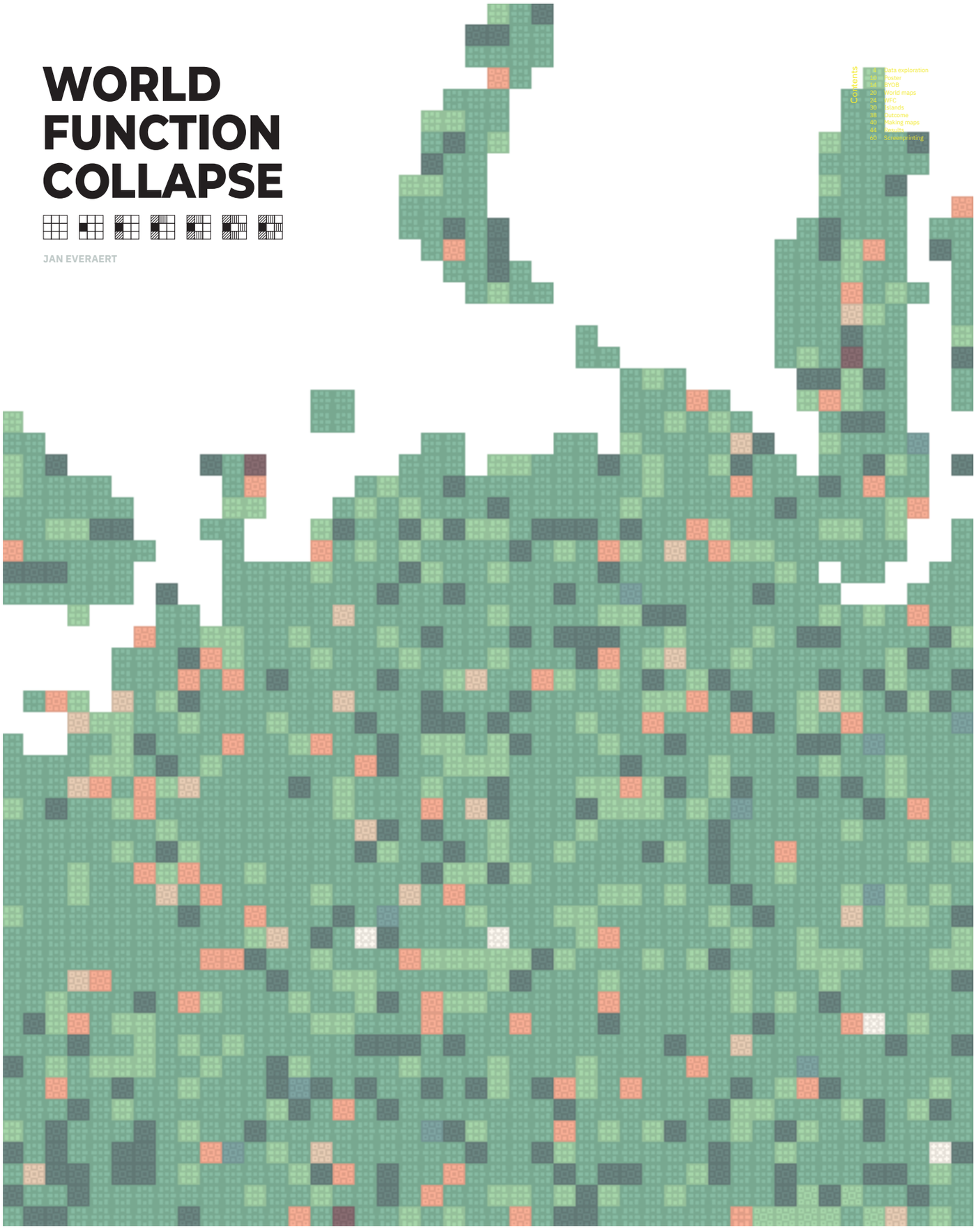
WORLD FUNCTION COLLAPSE

JAN EVERAERT

Contents

4 Data exploration
18 Poster
14 BYOB
20 World maps
24 WFC
30 Islands
38 Outcome
40 Faking maps
44 Textile
60 Screenprinting



Introduction

This magazine serves as an overview to the process that enabled the “world function collapse” project. The end result (aside from this magazine) is a map (72x102 centimeters) of the world. The data, captured by ESA (the European Space Agency) is embeded in the map, and screenprinted in UV reflective ink on the map. The user is enticed to explore this data using a magnifier with a little LED light embeded in it.
The process has gone through more than 50 iterations before reaching it’s (intermediate) conclusion.

September 2023

4 Data exploration

10 Data poster

14 Bring Your Own Beamer

20 World maps

24 Wave function collapse

30 Data islands

38 ESA competition

40 World function collapse

44 The (intermediate) result

December 2023

60 Screenprinting

The dataset

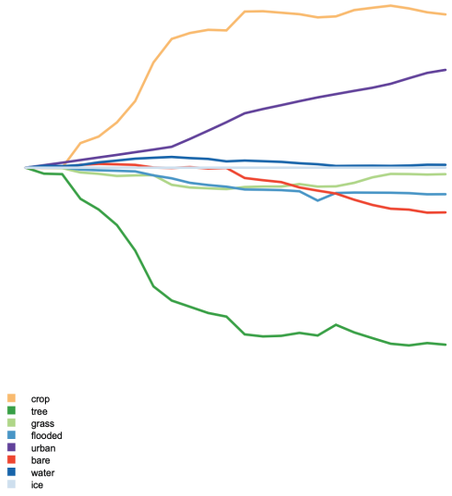
Every year, the European Space Agency, sends out a competition. The aim of this competition is to create a data visualization with a very clear climate related message/ This can be a visualization of trends, of variability over seasons, a subset of the data etc. A github is provided with a lot of satellite captured data, for example glacier extent, lakes, sea surface temperatures, greenhouse gasses, land surface temperature, arctic ice extent and others.

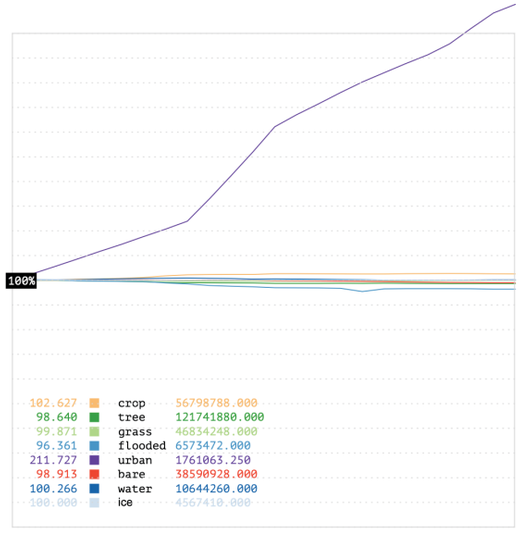
One of these datasets is the “land cover (LC-1)” dataset. For each country in the dataset, there is a table included noting the land cover composition between 1992 and 2015. This data is in km2, and based on the UN Land Cover Classification System (LCCS). In short: for every country, we can tell how much km2 was dedicated to grasslands, tree-cover, urban, etc, and are able to tell how it changed from year to year. The LCCS system uses two phases to create these classification files; the definition phase, and the modular phase (in reality, this is called the dichotomous phase and the modular-hierarchical phase).

In the first phase, eight major land cover types are defined: (1) Cultivated and Managed Terrestrial Areas, (2) Natural and Semi-Natural Terrestrial Vegetation, (3) Cultivated Aquatic or Regularly Flooded Areas, (4) Natural and Semi-Natural Aquatic or Regularly Flooded Vegetation, (5) Artificial Surfaces and Associated Areas, (6) Bare Areas, (7) Artificial Water bodies, Snow and Ice, and (8) Natural Water bodies, Snow and Ice. The second phase notes the combination of two or more of these categories. For example, you can have a mix of flooded and tree-covered areas, or a mix of grasslands and lower bush coverings.

To be complete; the tables had the headers on the right, which later got reduced and combined to more readable and understandable (on the right).

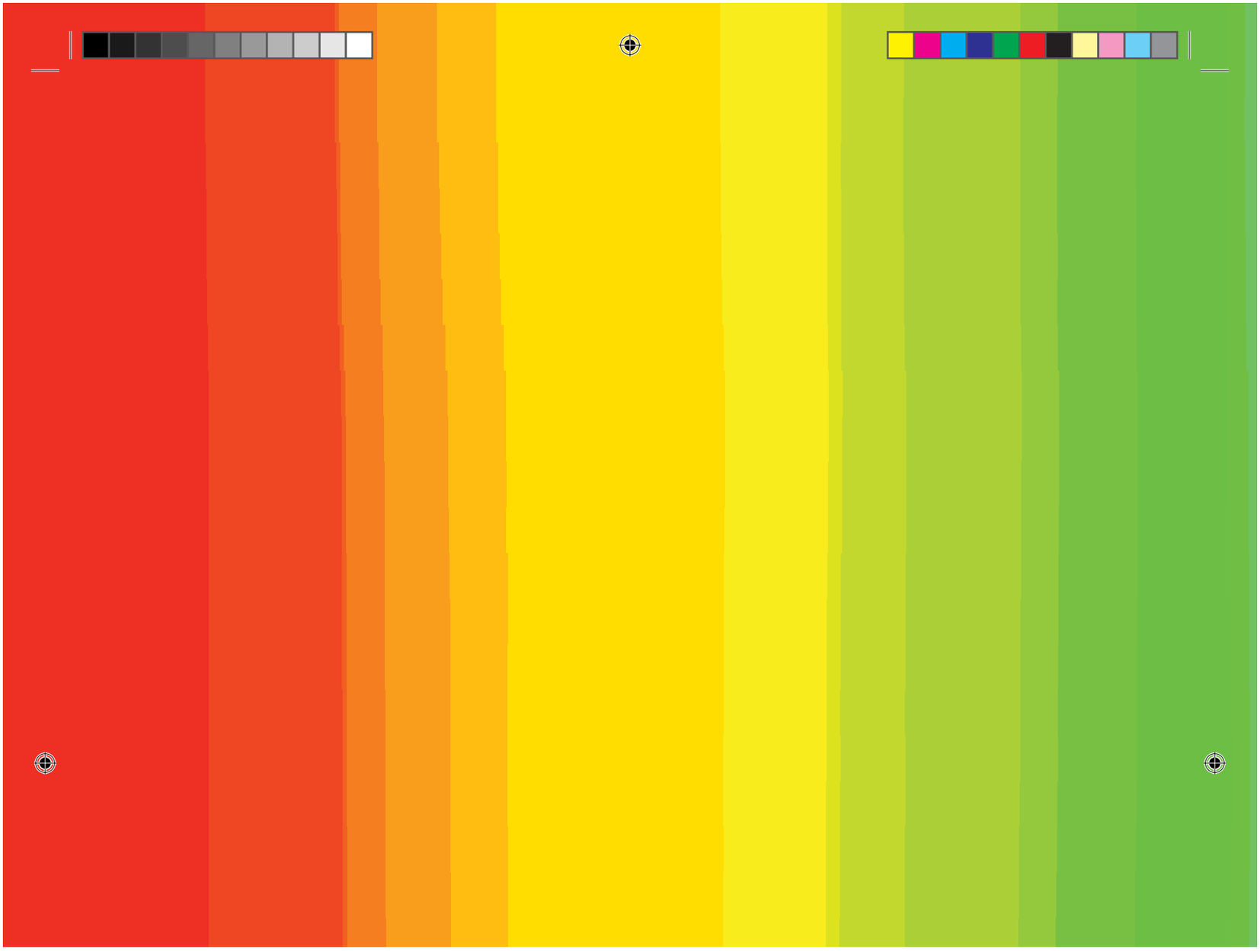
In order to better understand this dataset, I’ve visualized it in bands of color on the next page, as a stacked chart. This was meant to explore the limitations of the dataset, and to see how much the data changes over time.



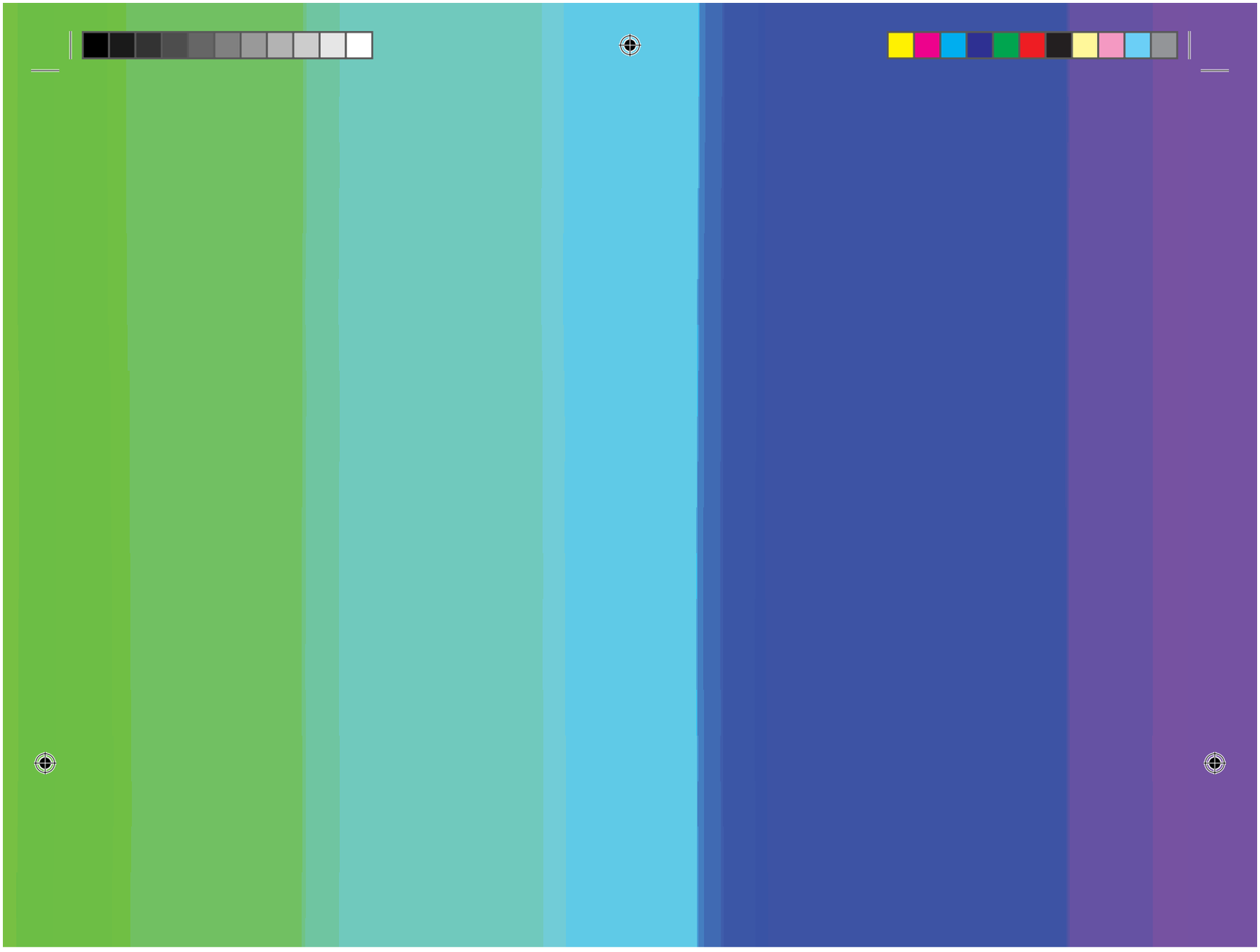


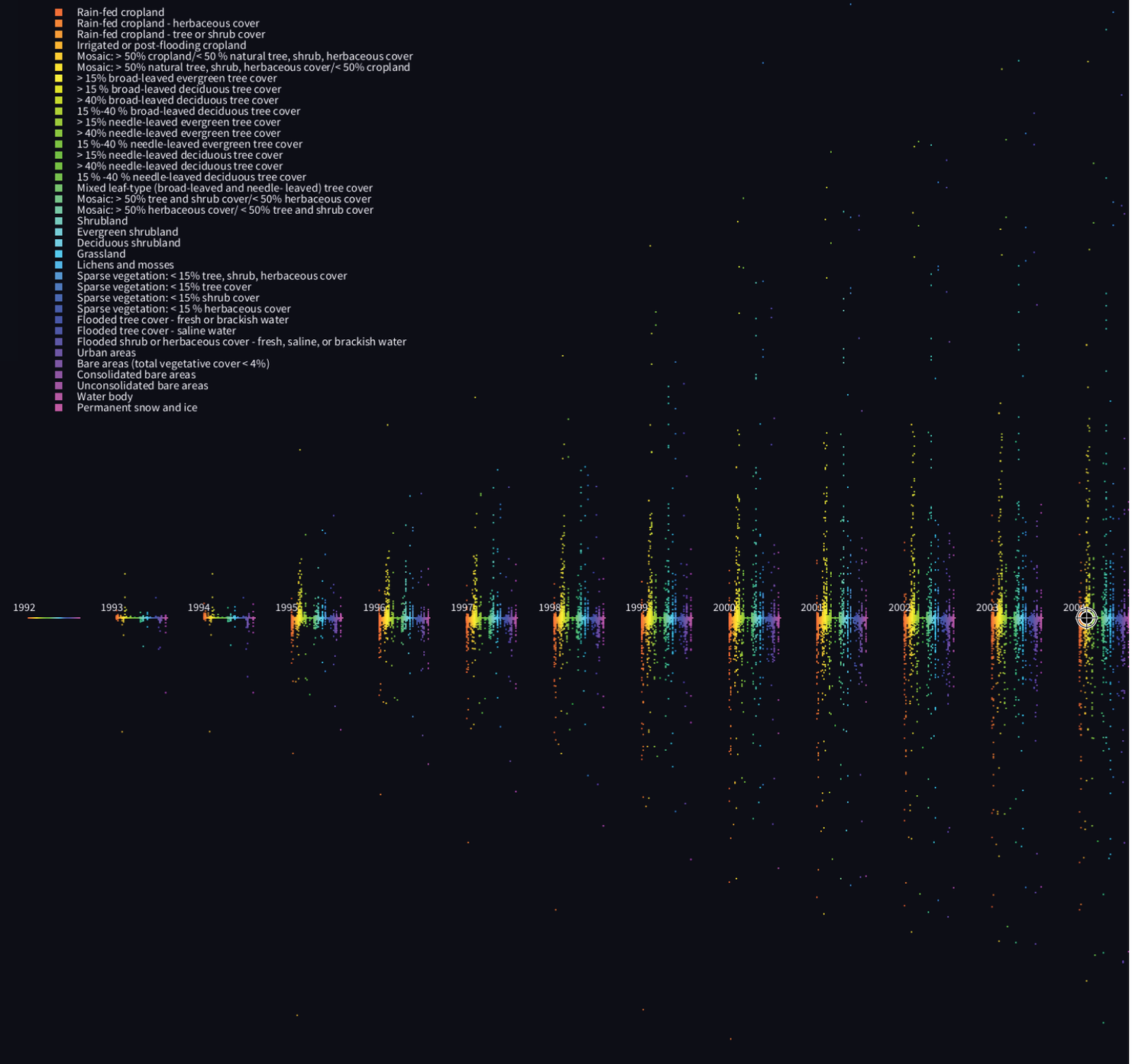
Here above are another exploration of the data. Both depicts the increase or decrease in data, but left does not take into account the distribution of the land type. Because of this, it looks like a lot of forest has been replaced by trees. This is a false derivation. We cannot learn this from this visualization.

Year	Year
Crop	Rain-fed cropland
	Rain-fed cropland - herbaceous cover
	Rain-fed cropland - tree or shrub cover
	Irrigated or post-flooding cropland
	Mosaic: > 50% cropland/< 50 % natural tree, shrub, herbaceous cover
	Mosaic: > 50% natural tree, shrub, herbaceous cover/< 50% cropland
Tree	> 15% broad-leaved evergreen tree cover
	> 15 % broad-leaved deciduous tree cover
	> 40% broad-leaved deciduous tree cover
	15 %-40 % broad-leaved deciduous tree cover
	> 15% needle-leaved evergreen tree cover
	> 40% needle-leaved evergreen tree cover
	15 %-40 % needle-leaved evergreen tree cover
	> 15% needle-leaved deciduous tree cover
	> 40% needle-leaved deciduous tree cover
	15 % -40 % needle-leaved deciduous tree cover
	Mixed leaf-type (broad-leaved and needle- leaved) tree cover
	Mosaic: > 50% tree and shrub cover/< 50% herbaceous cover
	Mosaic: > 50% herbaceous cover/ < 50% tree and shrub cover
	Shrubland
	Evergreen shrubland
	Deciduous shrubland
Grass	Grassland
	Lichens and mosses
	Sparse vegetation: < 15% tree, shrub, herbaceous cover
	Sparse vegetation: < 15% tree cover
	Sparse vegetation: < 15% shrub cover
	Sparse vegetation: < 15 % herbaceous cover
Flooded	Flooded tree cover - fresh or brackish water
	Flooded tree cover - saline water
	Flooded shrub or herbaceous cover - fresh, saline, or brackish water
Urban	Urban areas
Bare	Bare areas (total vegetative cover < 4%)
	Consolidated bare areas
	Unconsolidated bare areas
Water	Water body
Ice	Permanent snow and ice

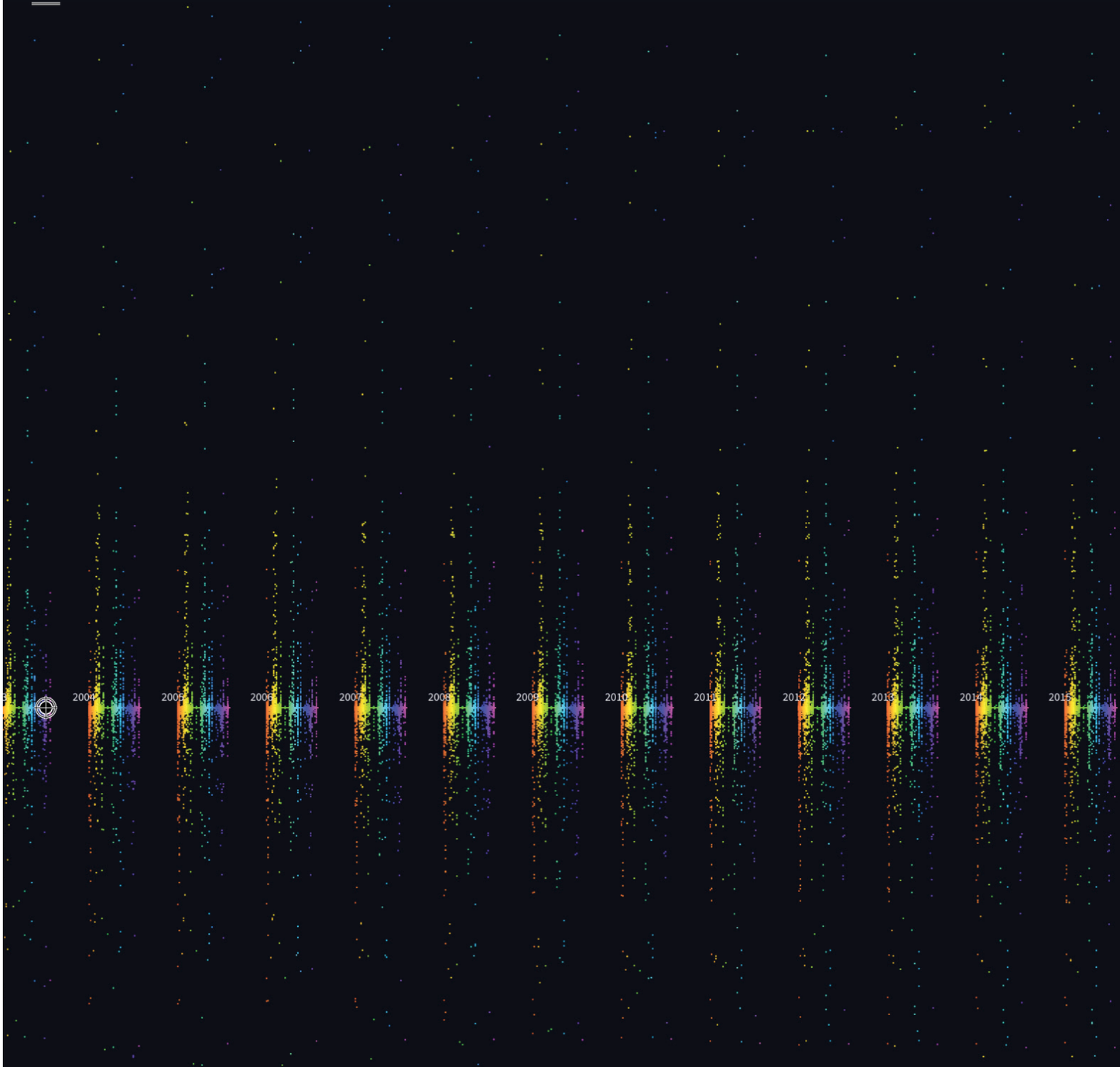


year
Rain-fed cropland
Rain-fed cropland - herbaceous cover
Rain-fed cropland - tree or shrub cover
Irrigated or post-flooding cropland
Mosaic: > 50% cropland/< 50 % natural tree, shrub, herbaceous cover
Mosaic: > 50% natural tree, shrub, herbaceous cover/< 50% cropland
> 15% broad-leaved evergreen tree cover
> 15 % broad-leaved deciduous tree cover
> 40% broad-leaved deciduous tree cover
15 %-40 % broad-leaved deciduous tree cover
> 15% needle-leaved evergreen tree cover
> 40% needle-leaved evergreen tree cover
15 %-40 % needle-leaved evergreen tree cover
> 15% needle-leaved deciduous tree cover
> 40% needle-leaved deciduous tree cover
15 % -40 % needle-leaved deciduous tree cover
Mixed leaf-type (broad-leaved and needle- leaved) tree cover
Mosaic: > 50% tree and shrub cover/< 50% herbaceous cover
Mosaic: > 50% herbaceous cover/ < 50% tree and shrub cover
Shrubland
Evergreen shrubland
Deciduous shrubland
Grassland
Lichens and mosses
Sparse vegetation: < 15% tree, shrub, herbaceous cover
Sparse vegetation: < 15% tree cover
Sparse vegetation: < 15% shrub cover
Sparse vegetation: < 15 % herbaceous cover
Flooded tree cover - fresh or brackish water
Flooded tree cover - saline water
Flooded shrub or herbaceous cover - fresh, saline, or brackish water
Urban areas
Bare areas (total vegetative cover < 4%)
Consolidated bare areas
Unconsolidated bare areas
Water body
Permanent snow and ice

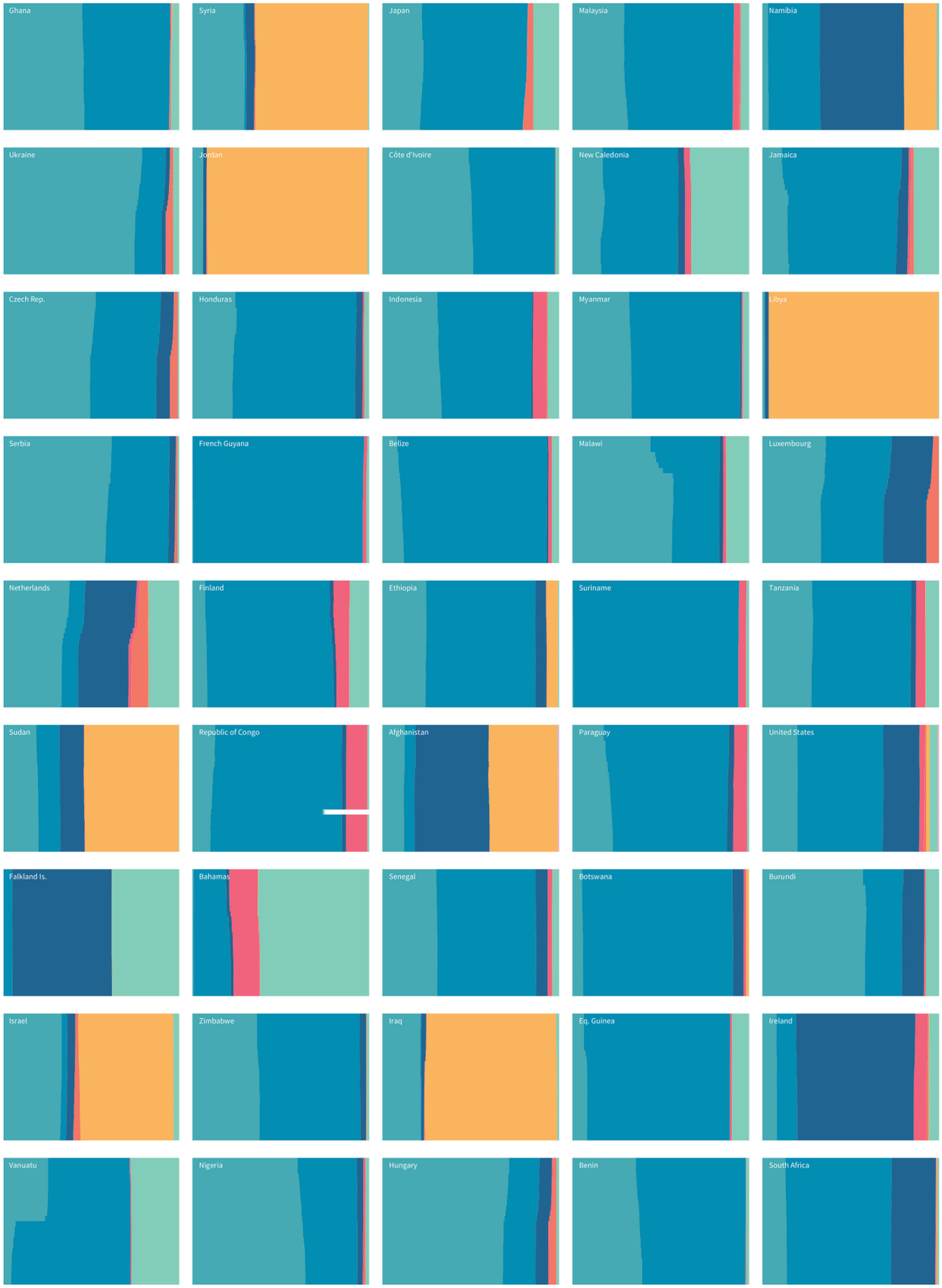


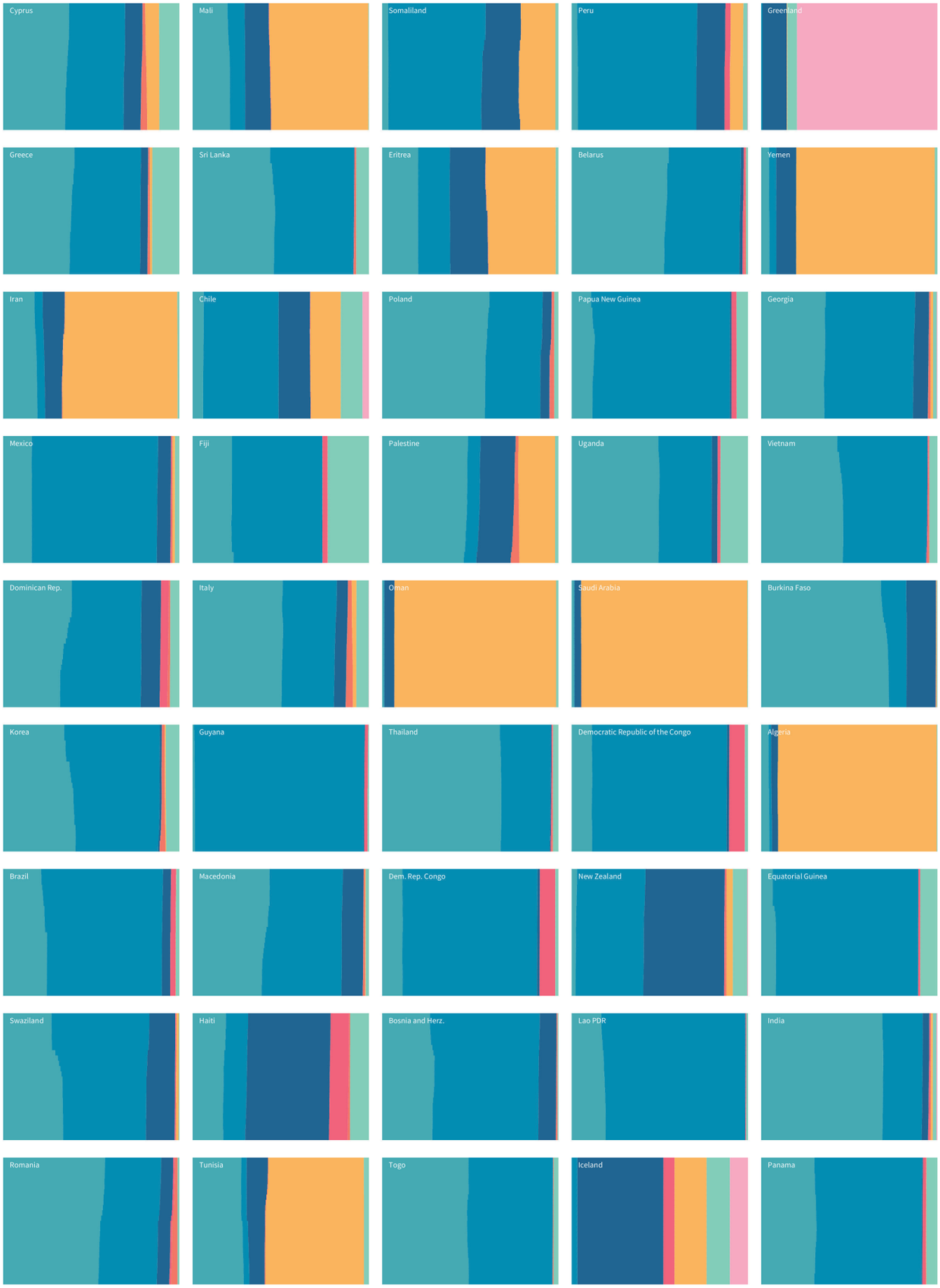


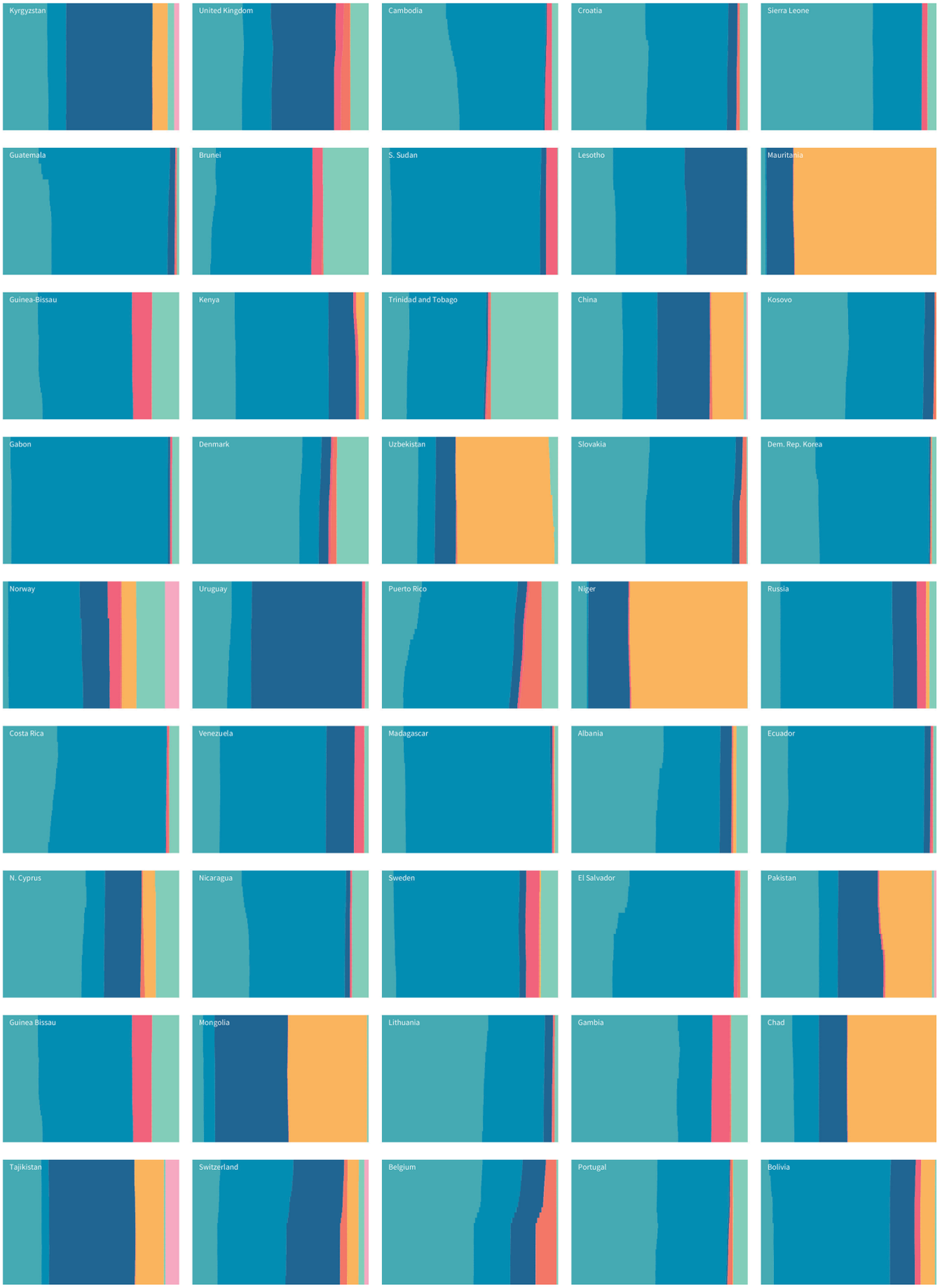
This chart displays the variation over time, starting at 100%. It was an experiment, trying to display all the data at once, in a singular visual. This gives a vague idea in trends, but is hardly readable as the countries of origin are indistinguishable.

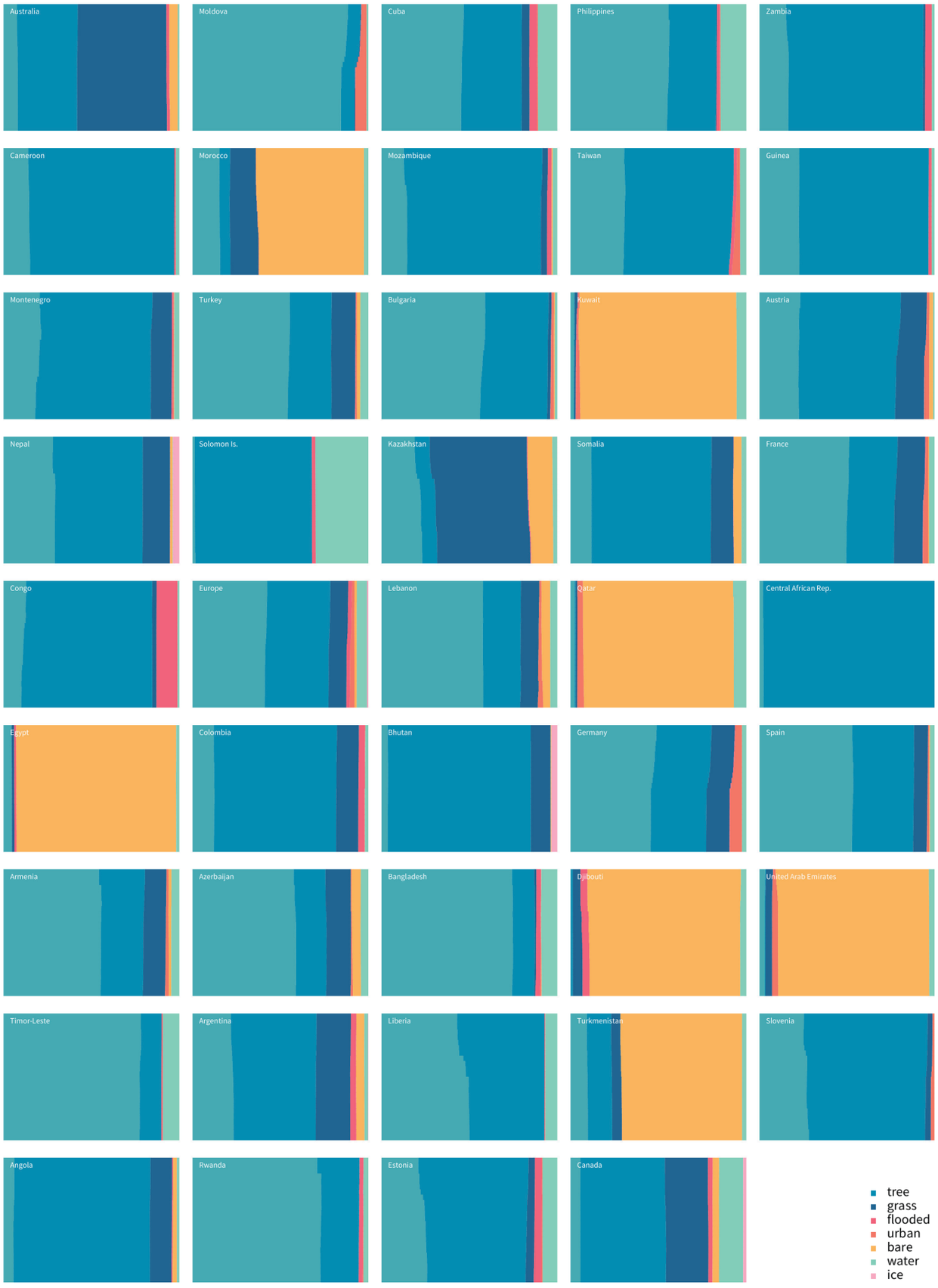


The following pages are an alteration on the visualization of stacked charts per country. Every country is plotted on the poster, using the same color code. Gathering all the countries in a single visual serves to get insight into the composition of land over the entire world.









BYOB

During the months I've worked on this dataset, an event occurred in Ghent, called "Bring your own beamer", organized by Nerdlab VZW. Together with a friend and partner, Dennis Baptist. we've used the dataset to create an exhibition piece. The piece consisted of a generated, animated graphical representation of the composition of land, categorized in smaller subsets for readability. Throughout the 10 seconds a country was shown, the visual was altered to show the composition at a specific year.

We've experimented with a few forms, in an effort to optimize for projection (to keep the contrast high). Color schemes we're selected rather for visual effect than for accurate representation of the data.

The visual piece was accompanied by a generative audio piece, composed by Dennis. Using Ableton as a soundboard, midi was sent from the visual generative program to Ableton to translate the data into sound.

From the website "byobworldwide.com":

BYOB is a series of one-night-exhibitions curated by different people around the world. The idea is simple:

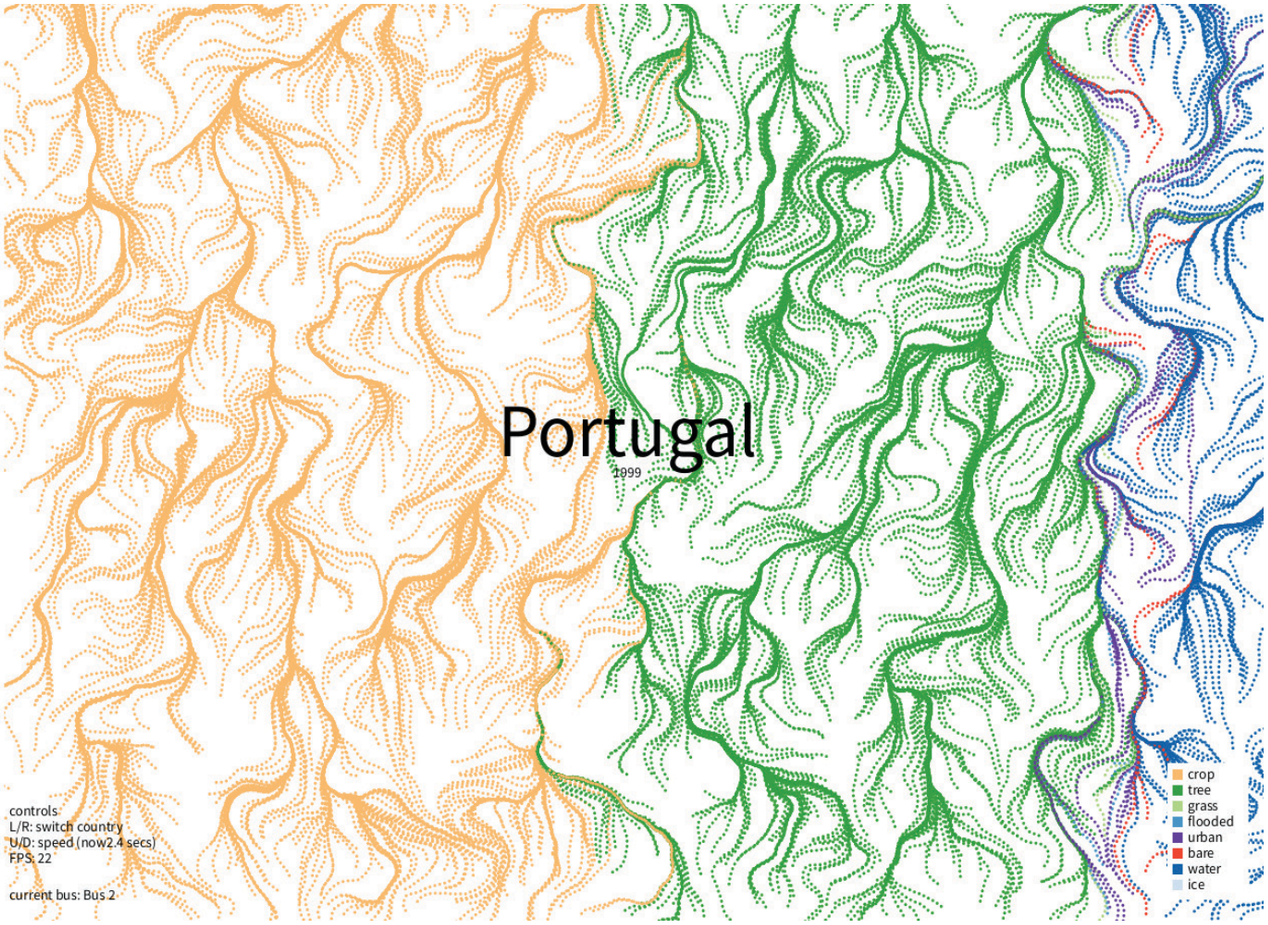
--> Find a place

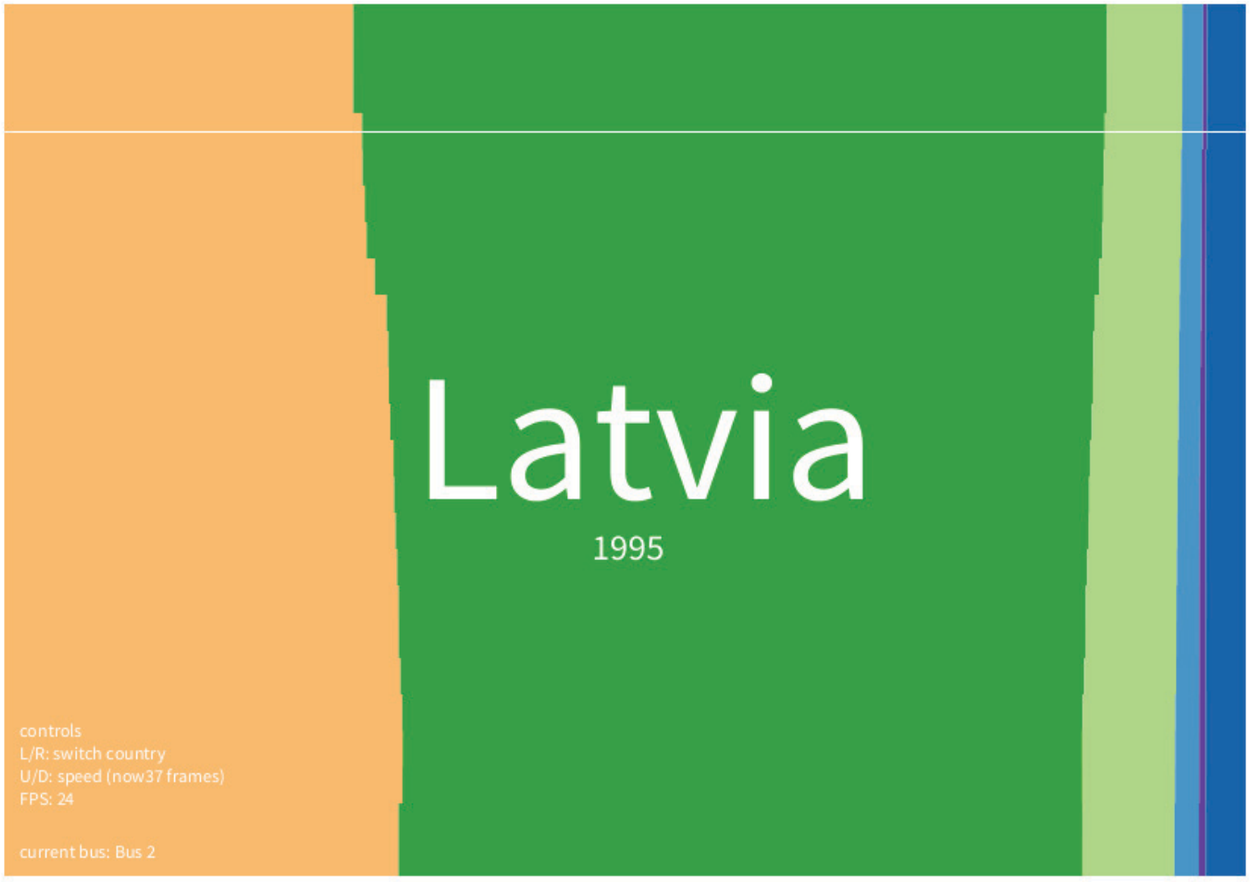
--> Invite many artists

--> Ask them to bring their projectors

BYOB is a way of making a big exhibition with zero budget. It is an exploration of the medium of projection.

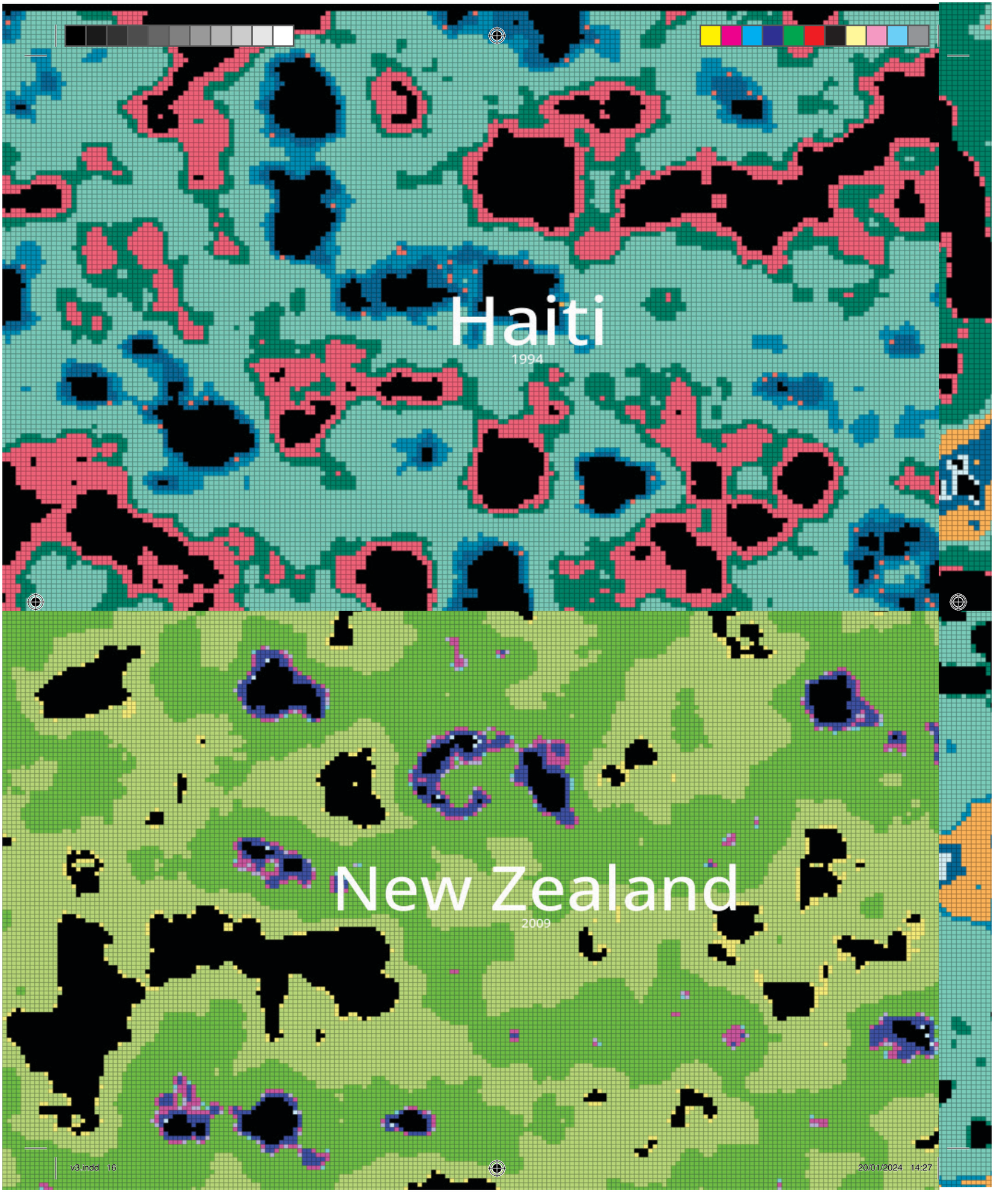
Everyone is allowed to create a non-commercial event, and they happen all over the world.

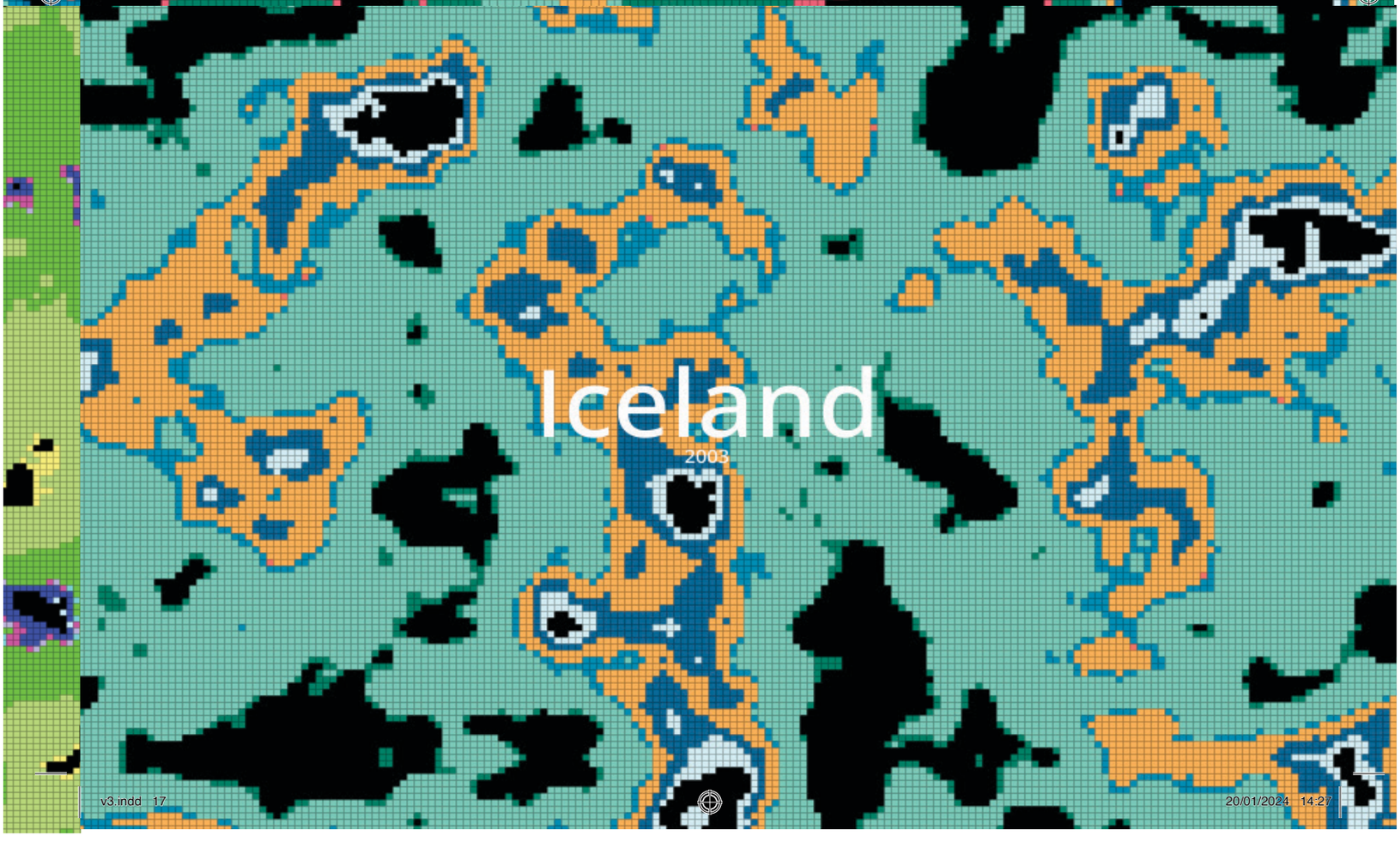


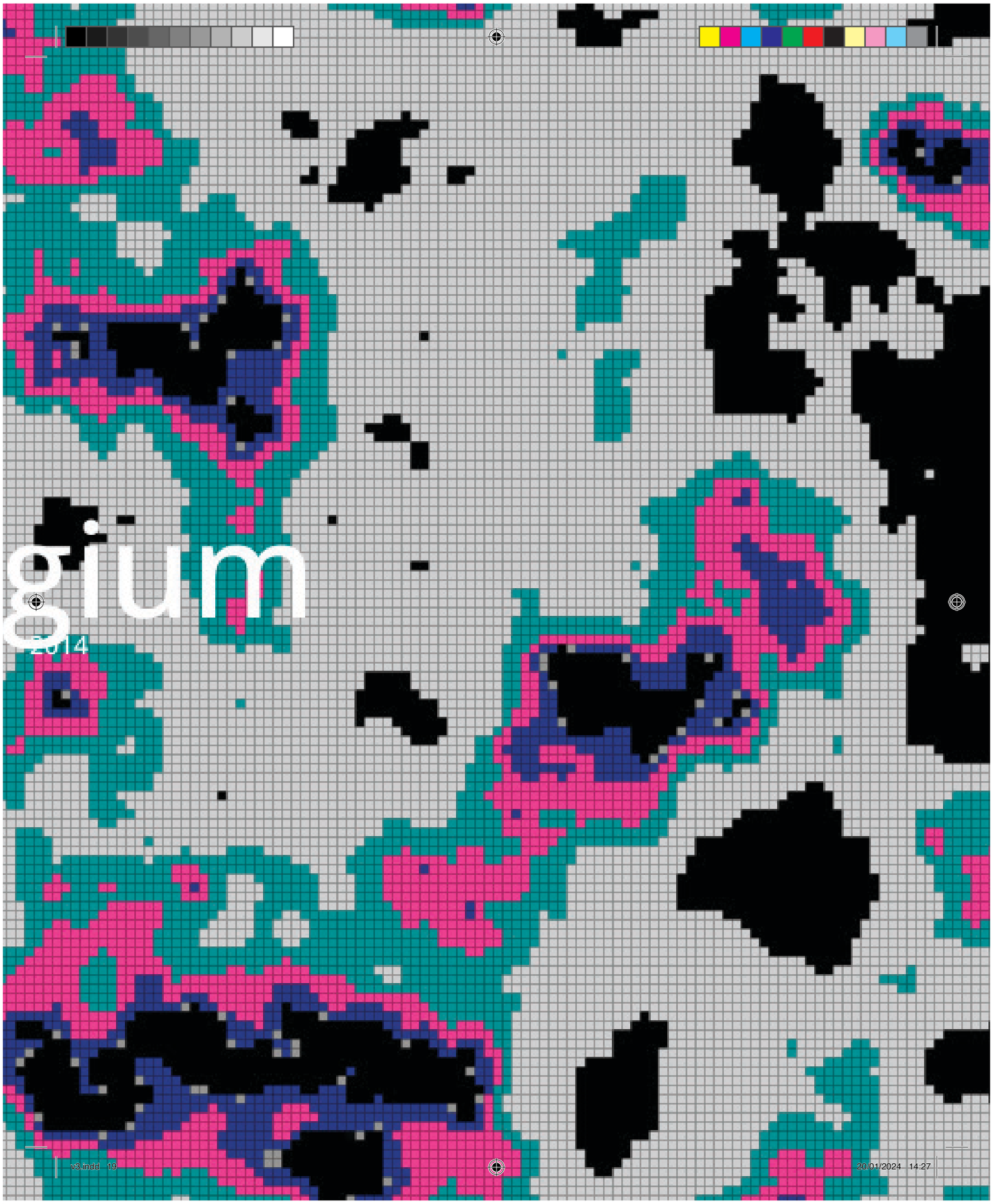


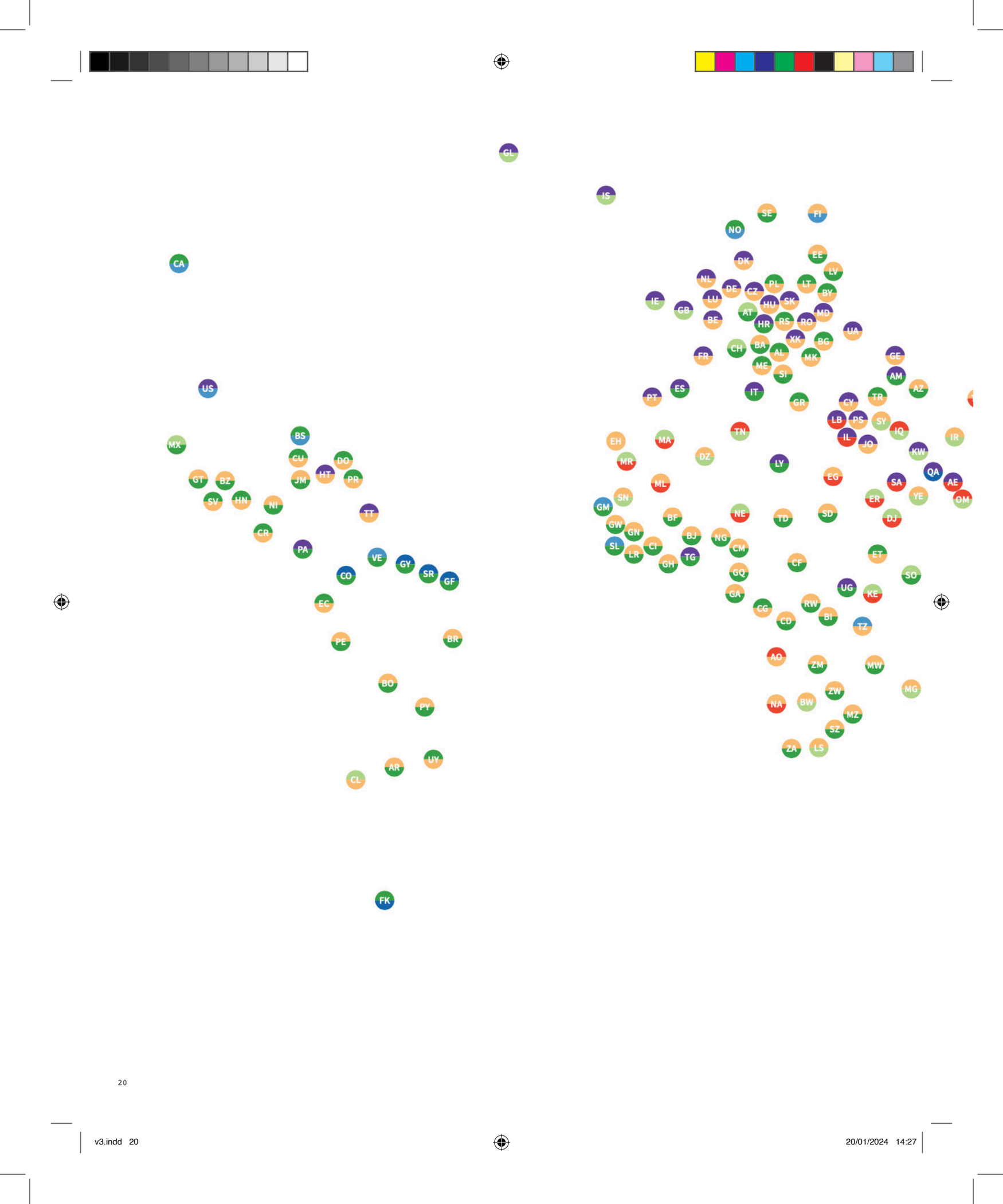
Eventually we chose to use a noise map as a base. Using a tile grid assigned to this noise, every tile had a value between 0 and 1. When a country was shown, we iterated over the tiles, and used the data to color it into a specific category. When a country had for example 40% crops (the first value in the list), values up until 0.4 were colored corresponding to the crop category. The following pages show a few examples of how this looked at the time of exhibition. For every change in country, we selected a new color scheme, and shifted momentarily to a visualization consisting of bars, showing the same spread of colors on a linear axis.

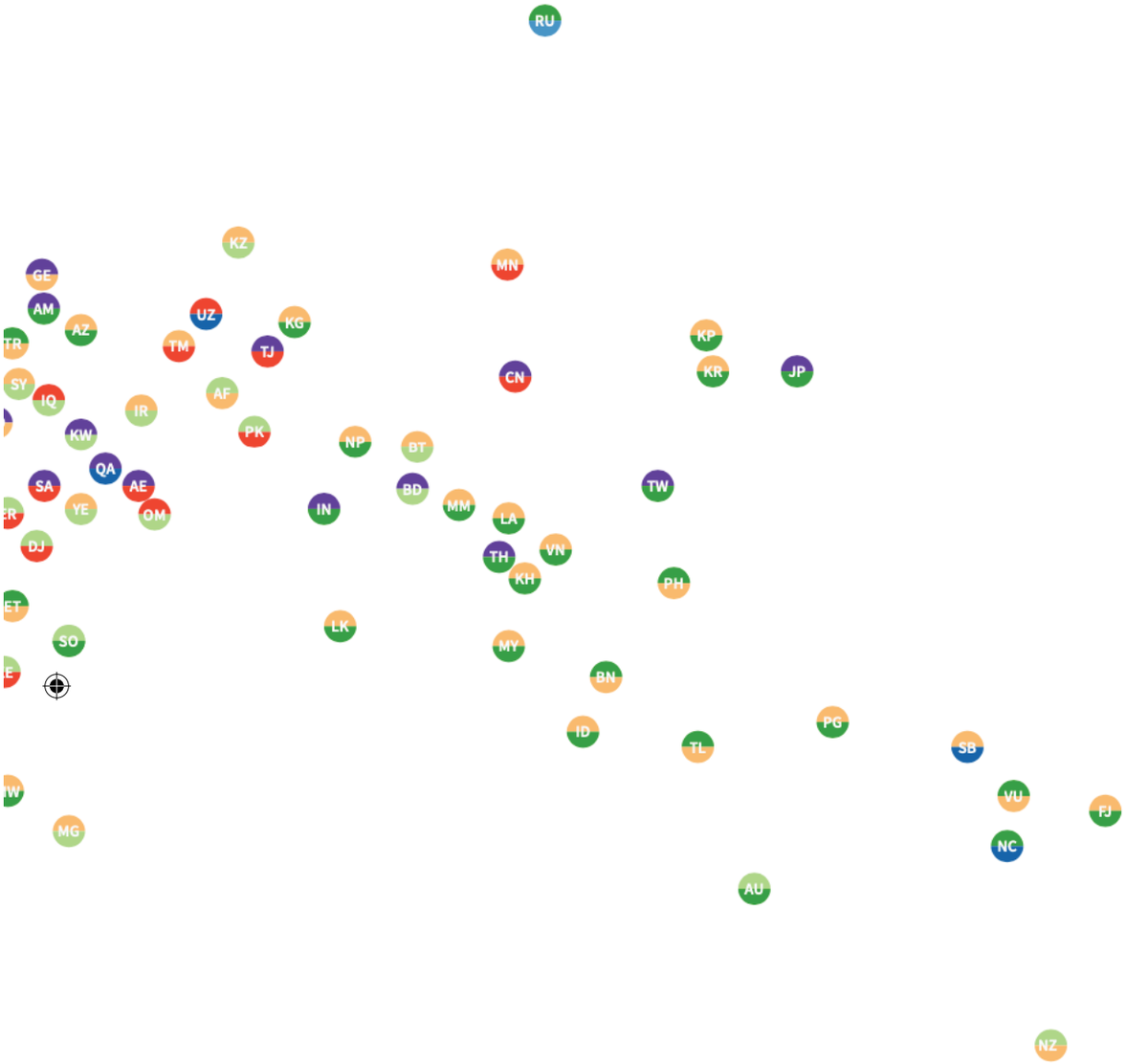
During the evening, we had interested visitors asking questions about the data, and standing by to see the differences in the visuals between countries and over time.











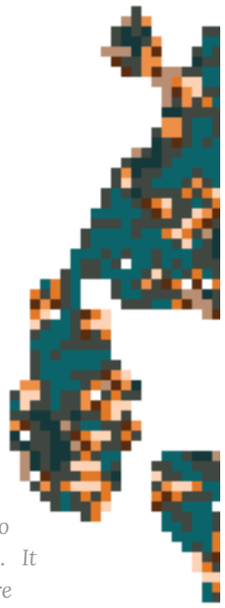
During the explorations, I was experimenting with visualizing the most changing fields per country. the next page consists of a visualization in a stock market style, displaying each country and the corresponding land composition fields that have risen most since 1992, and the one that has gone down the most. The map above does the same, positioning the dot as close to the position of the country's coordinates, keeping readability in mind. Both share the same legend

- crop
- tree
- grass
- flooded
- urban
- bare
- water
- ice

AF	Afghanistan	AL	Albania	DZ	Algeria
AM	Armenia	AU	Australia	AT	Austria
BD	Bangladesh	BY	Belarus	BE	Belgium
BT	Bhutan	BO	Bolivia	BA	Bosnia and Herzegovi
BN	Brunei	BG	Bulgaria	BF	Burkina Faso
CM	Cameroon	CA	Canada	CF	Central African Repu
CN	China	CO	Colombia	CG	Congo [Republic]
HR	Croatia	CU	Cuba	CY	Cyprus
CD	Congo [DRC]	DK	Denmark	DJ	Djibouti
EG	Egypt	SV	El Salvador	GN	Guinea
EE	Estonia	ET	Ethiopia	FK	Falkland Islands
FR	France	GF	French Guiana	GA	Gabon
DE	Germany	GH	Ghana	GR	Greece
GW	Guinea-Bissau	GN	Guinea	GW	Guinea-Bissau
HN	Honduras	HU	Hungary	IS	Iceland
IR	Iran	IQ	Iraq	IE	Ireland
JM	Jamaica	JP	Japan	JO	Jordan
KR	South Korea	XK	Kosovo	KW	Kuwait
LV	Latvia	LB	Lebanon	LS	Lesotho
LT	Lithuania	LU	Luxembourg	MK	Macedonia [FYROM]
MY	Malaysia	ML	Mali	MR	Mauritania
MN	Mongolia	ME	Montenegro	MA	Morocco
CY	Cyprus	NA	Namibia	NP	Nepal
NZ	New Zealand	NI	Nicaragua	NE	Niger
OM	Oman	PK	Pakistan	PS	Palestinian Territor
PY	Paraguay	PE	Peru	PH	Philippines
PR	Puerto Rico	QA	Qatar	RO	Romania
SD	Sudan	SA	Saudi Arabia	SN	Senegal
SK	Slovakia	SI	Slovenia	SB	Solomon Islands
ZA	South Africa	ES	Spain	LK	Sri Lanka
SZ	Swaziland	SE	Sweden	CH	Switzerland
TJ	Tajikistan	TZ	Tanzania	TH	Thailand
TT	Trinidad and Tobago	TN	Tunisia	TR	Turkey
UA	Ukraine	AE	United Arab Emirates	GB	United Kingdom
UZ	Uzbekistan	VU	Vanuatu	VE	Venezuela
YE	Yemen	ZM	Zambia	ZW	Zimbabwe

	AO Angola	AR Argentina
	AZ Azerbaijan	BS Bahamas
	BZ Belize	BJ Benin
and Herzegovina	BW Botswana	BR Brazil
Faso	BI Burundi	KH Cambodia
African Republic	TD Chad	CL Chile
Republic]	CR Costa Rica	CI Côte d'Ivoire
	CZ Czech Republic	KP North Korea
i	DO Dominican Republic	EC Ecuador
	GQ Equatorial Guinea	ER Eritrea
d Islands	FJ Fiji	FI Finland
	GM Gambia	GE Georgia
	GL Greenland	GT Guatemala
Bissau	GY Guyana	HT Haiti
	IN India	ID Indonesia
	IL Israel	IT Italy
	KZ Kazakhstan	KE Kenya
	KG Kyrgyzstan	LA Laos
	LR Liberia	LY Libya
ia [FYROM]	MG Madagascar	MW Malawi
nia	MX Mexico	MD Moldova
	MZ Mozambique	MM Myanmar [Burma]
	NL Netherlands	NC New Caledonia
	NG Nigeria	NO Norway
nian Territories	PA Panama	PG Papua New Guinea
ines	PL Poland	PT Portugal
	RU Russia	RW Rwanda
	RS Serbia	SL Sierra Leone
Islands	SO Somalia	SO Somalia
ka	SD Sudan	SR Suriname
Land	SY Syria	TW Taiwan
d	TL Timor-Leste	TG Togo
	TM Turkmenistan	UG Uganda
Kingdom	US United States	UY Uruguay
La	VN Vietnam	
e	EH Western Sahara	

At this point, a new system came into view, called “wave function collapse”. In order to generate islands that corresponded in terms of composition, I needed to code a small piece of software to divide the tiles of an island into the relative fields. During the research phase, I’ve encountered this system as a way to dictate which tiles are placed together, and to prevent tiles that should not be close to be placed as such. It is often used in game development as a way to generate maps consistently and without weird placement like water next to a house when there should be a river bedding instead.



Wave function collapse

The Wave Function Collapse (WFC) algorithm is a technique used for procedural generation, primarily in game development. It allows for the creation of complex, structured patterns and landscapes that appear random but follow certain rules and constraints.

At its core, WFC operates on a grid of cells, where each cell can take on one of several predefined states. The goal is to generate a pattern or layout in this grid that adheres to given input constraints or sample data. The algorithm proceeds through the following steps (also see bellow):

1. Initialization: The grid is initialized with all possible states in each cell. One cell (chosen at random) is “collapsed”, brought back to a defined state.

2. Propagation: The algorithm then iterates through the grid, examining each cell and its neighboring cells. It looks at the patterns formed by neighboring cells and checks them against the input constraints or sample data. Cells that do not conform to these constraints are “collapsed” by removing the incompatible states.

3. Observation: The cells are sorted based on their number of possibilities, from low to high.

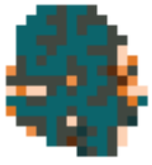
4. Superposition: The algorithm selects one (or multiple) cell from the start of the list and assigns it one of its remaining states, taking into account its neighbors and the constraints. This process continues until all cells have a single state.

The success of the WFC algorithm lies in its ability to create structured, realistic-looking patterns or layouts based on limited input data. It is widely used in procedural generation for tasks such as generating terrain, textures, mazes, and more.

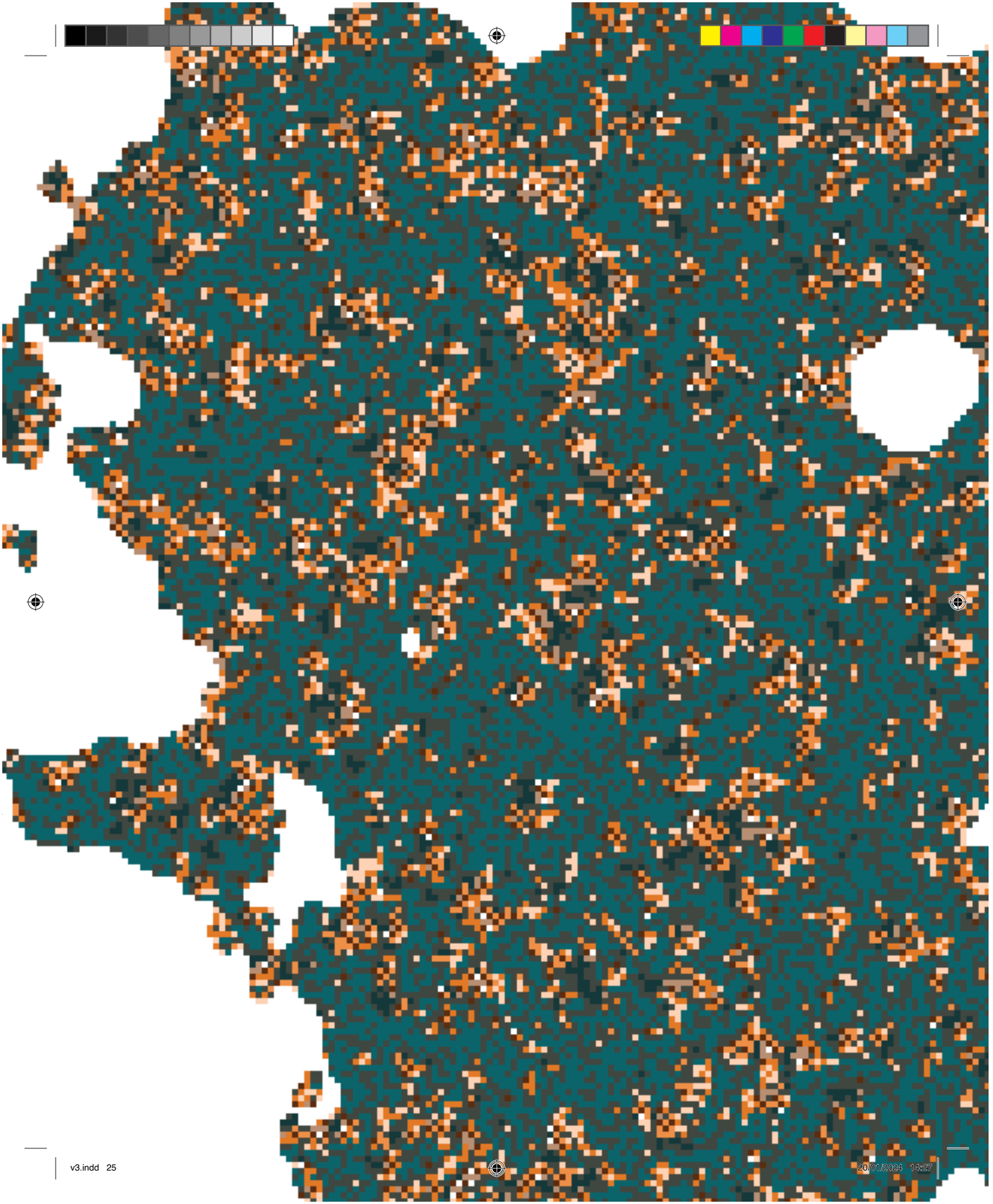
The next pages show this process step by step, collapsing over and over again.

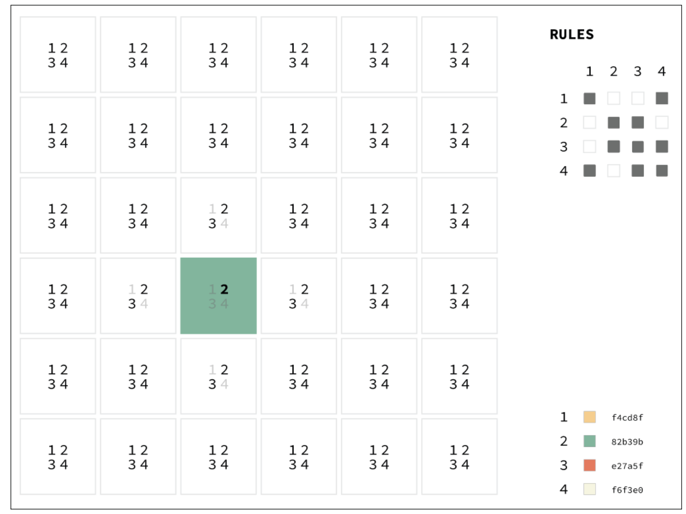
Wave Function Collapse can also be found in quantum mechanics. It represents a phenomenon where particles, such as electrons and photons, can exhibit both wave-like and particle-like properties under certain conditions. This duality is encapsulated by the Schrödinger equation, which describes the evolution of a quantum system as a wave function. The wave function contains all the information about a particle’s possible states and their probabilities.

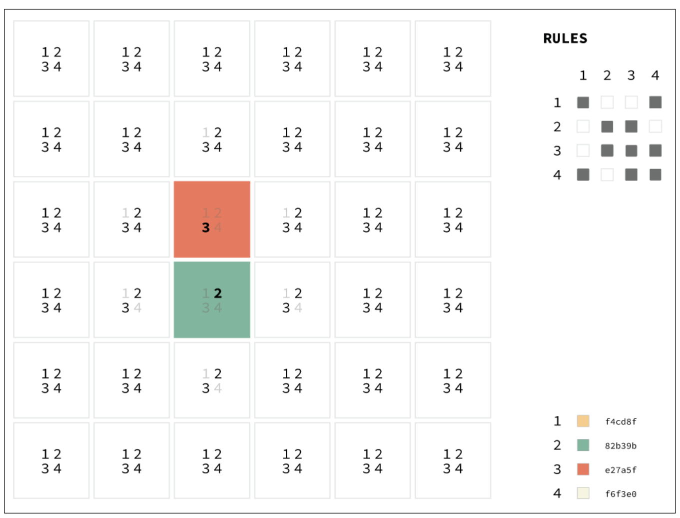
However, when we make a measurement in the quantum world, something intriguing happens. The wave function seems to “collapse” into a specific state. This means that the superposition of possible states instantly reduces to a single, well-defined outcome. The exact mechanism and interpretation of this collapse have been subjects of profound debate among physicists for many years.

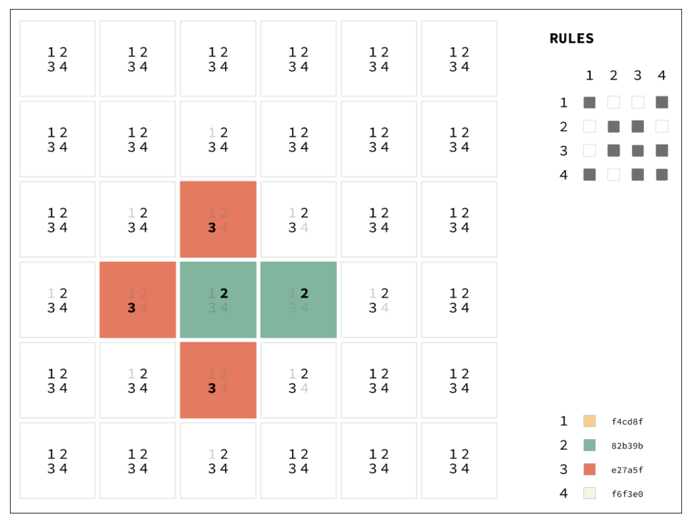


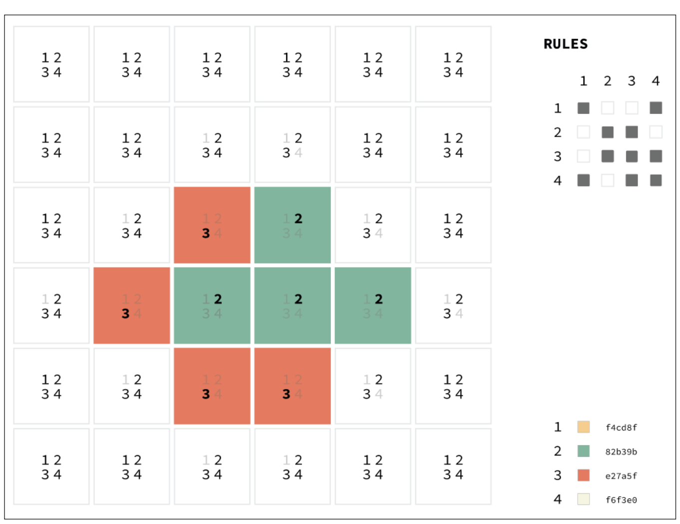
More information:
https://janeveraert.be/works/wfc/

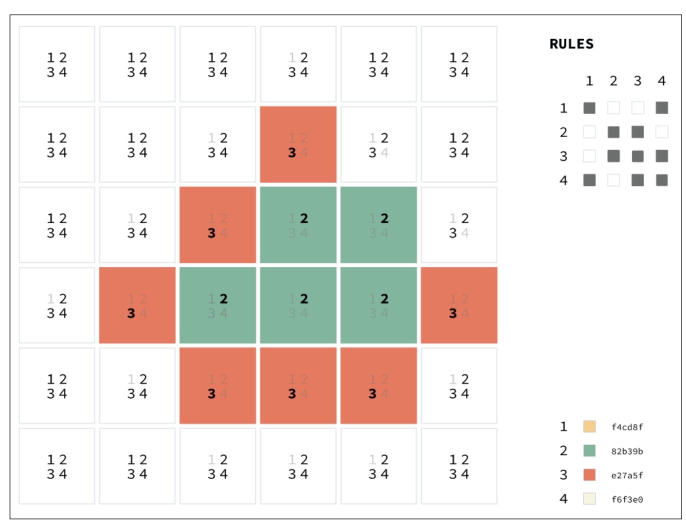


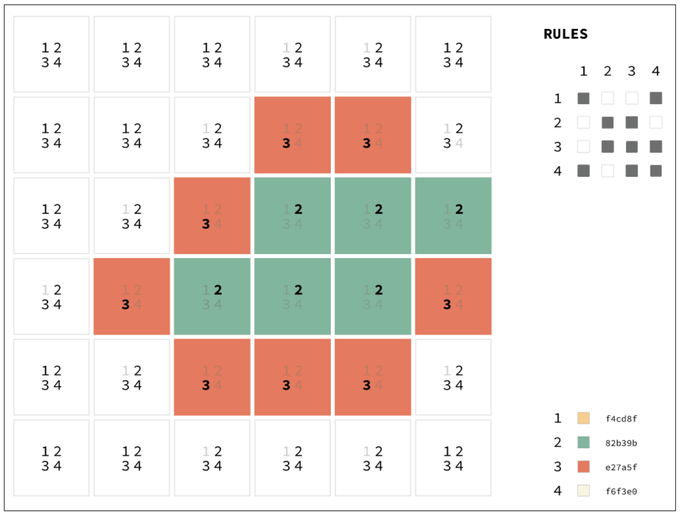


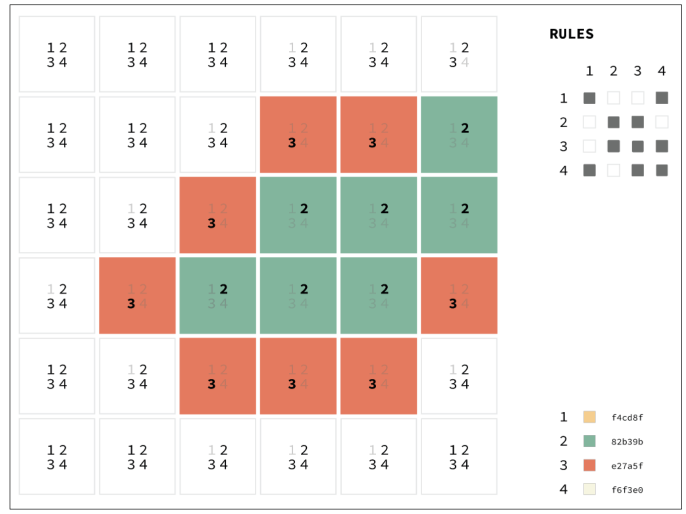


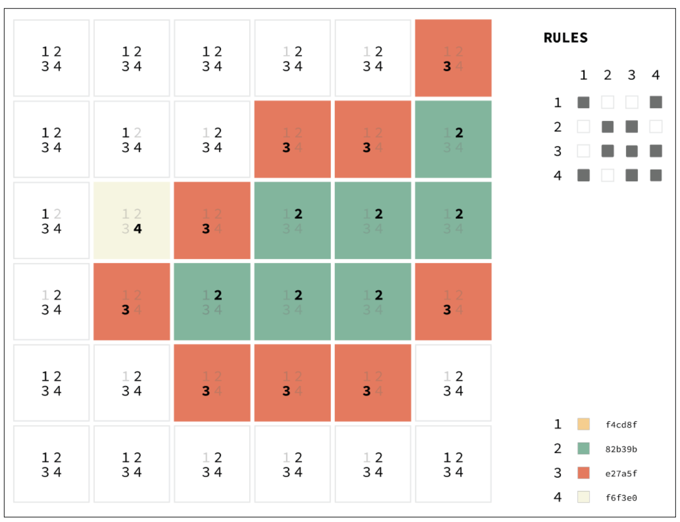


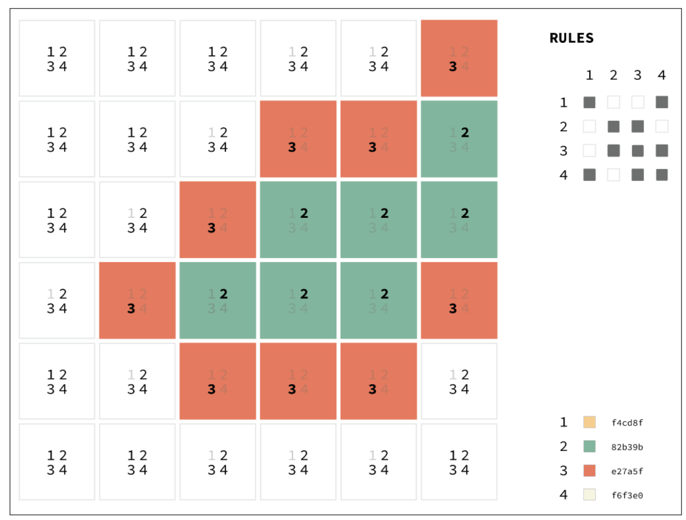


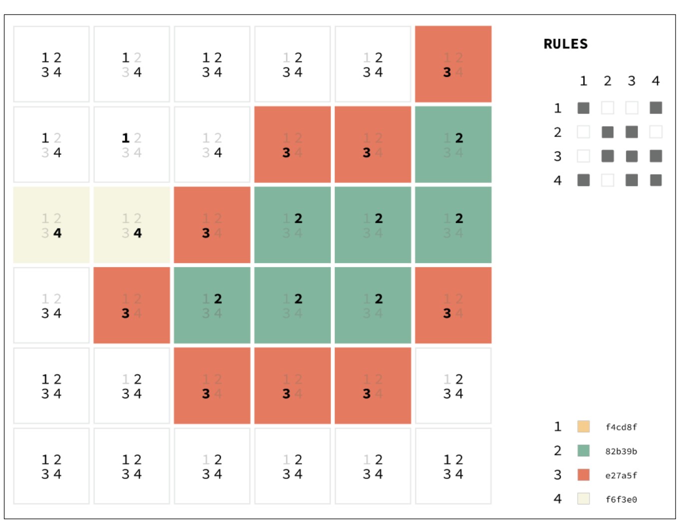


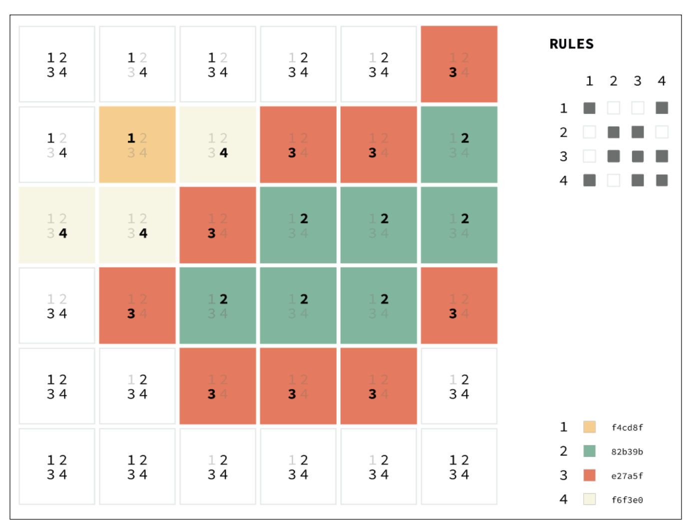


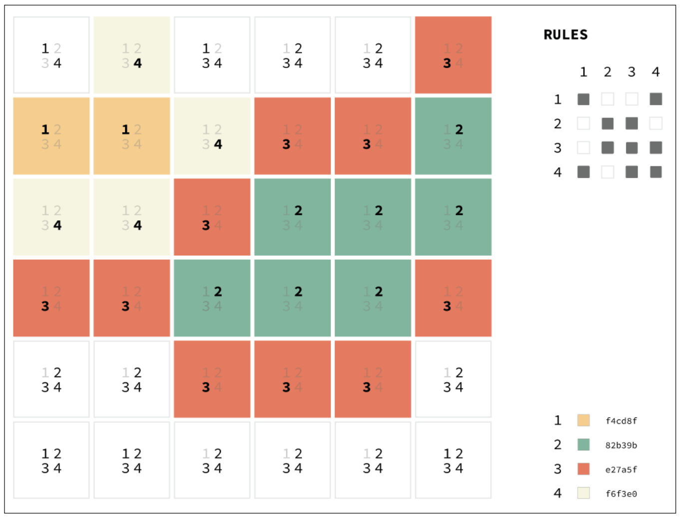


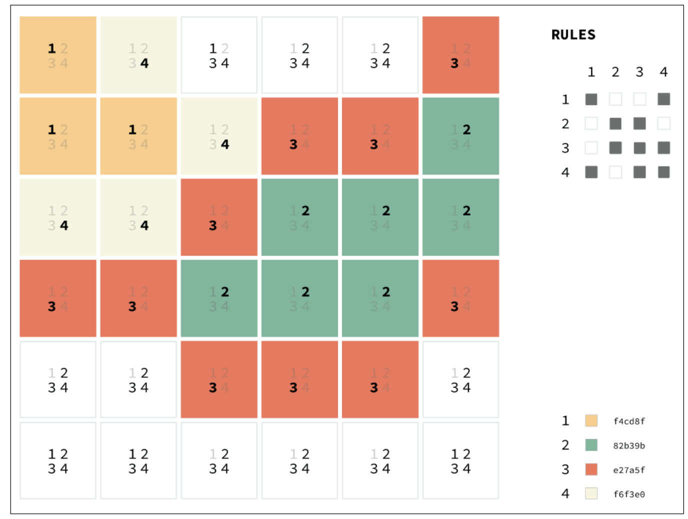


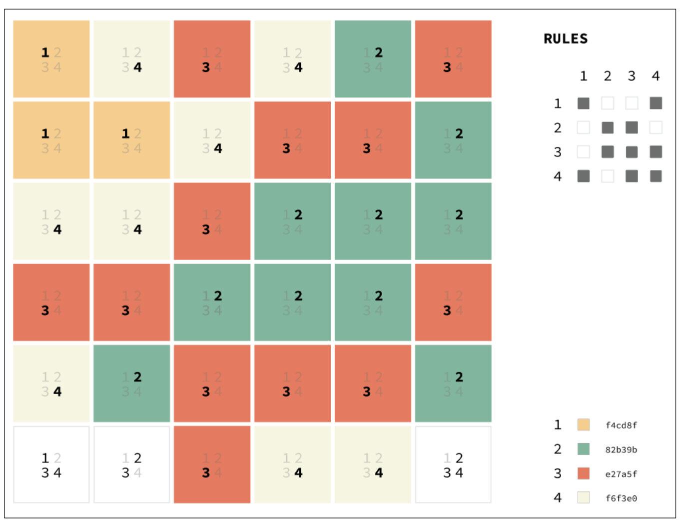


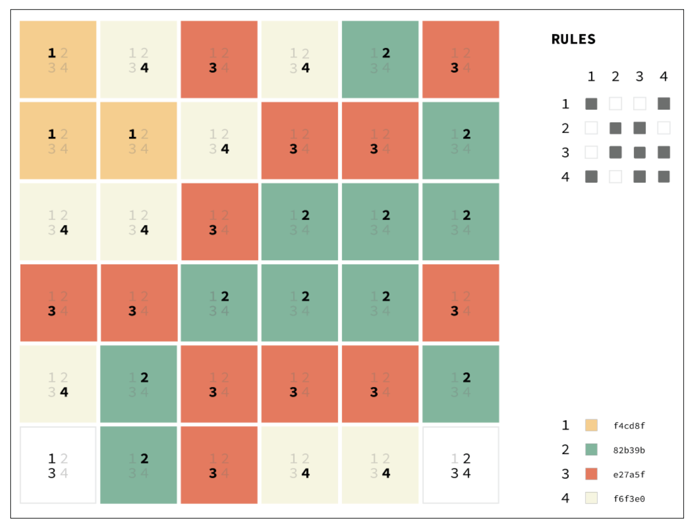


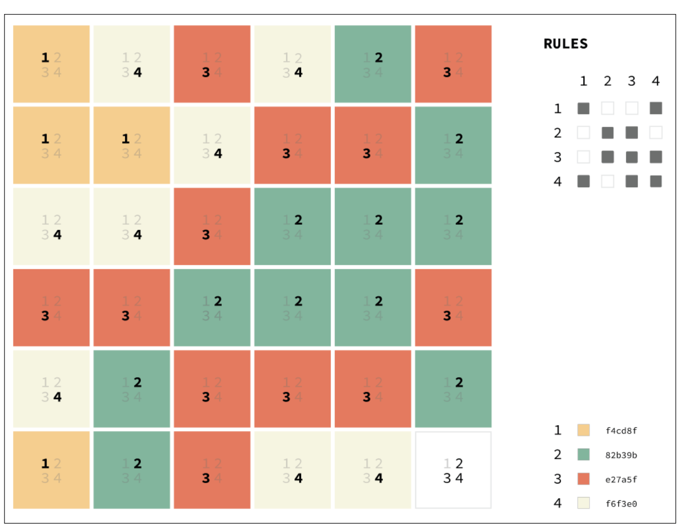


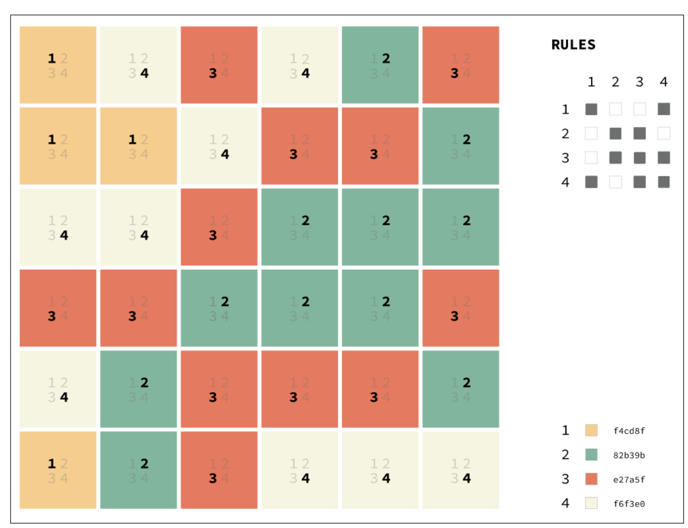

RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0


RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0


RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0


RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0


RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0


RULES
1 f4cd8f
2 82b39b
3 e27a5f
4 f6f3e0

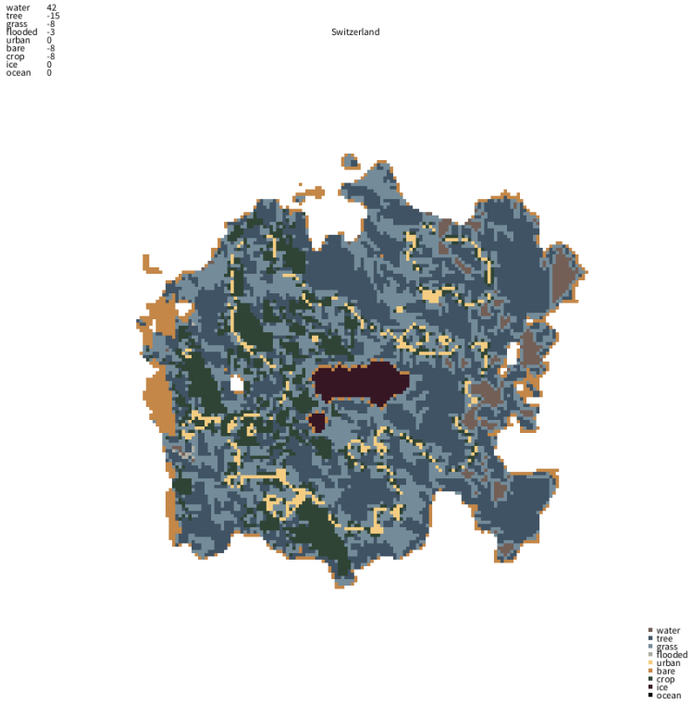
Manual approach

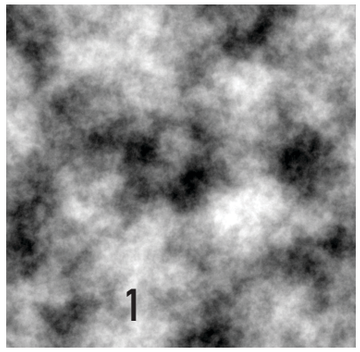
This system is quite versatile, and once it is implemented, you can package it and reuse it in different ways.

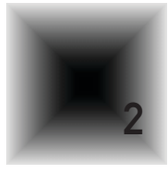
If the process is still unclear, try a more manual approach; Draw a grid on paper, grab an extra note and set up a few rules. These rules can for example note which colors are allowed next to each other. Start with coloring one random square, check the fields next to it and eliminate the colors it can't have based on the rules. Find the square with the lowest possible options (if more are available, pick one at random) and pick one of the options left (this is the "collapse" part). Once collapsed, update the squares next to it based on the rules (striking out the colors it can't have). Keep iterating over these steps until the page is full, or you're stuck. **Select a square with the least options, collapse it onto one final choice, and update the surrounding squares.**

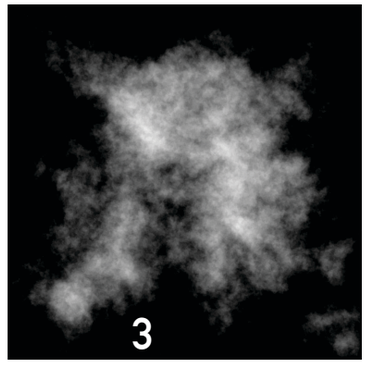
Some of the systems used in generative game environments take into account the surrounding fields, and the ones adjacent to those (going out two squares from the center). In this way, it's possible to prevent certain repetitive patterns, but makes the system more restrictive.

Even squares are not a requirement. **Townscaper.** a game by Oskar Stålberg and Raw Fury, makes use of pentagon tiles of varying size and orientation to keep the town you're building looking natural. This game is based largely on wave function collapse, and the only objective for a player is to create a town by selecting fields to place a house. The entire village is then updated to accomodate the change (using WFC).







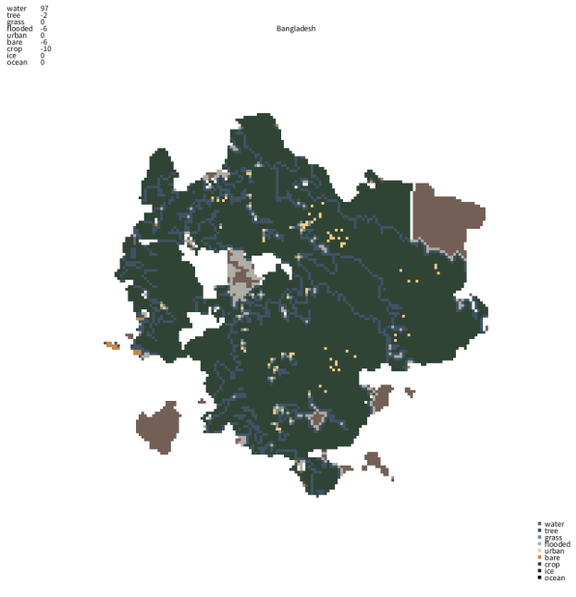


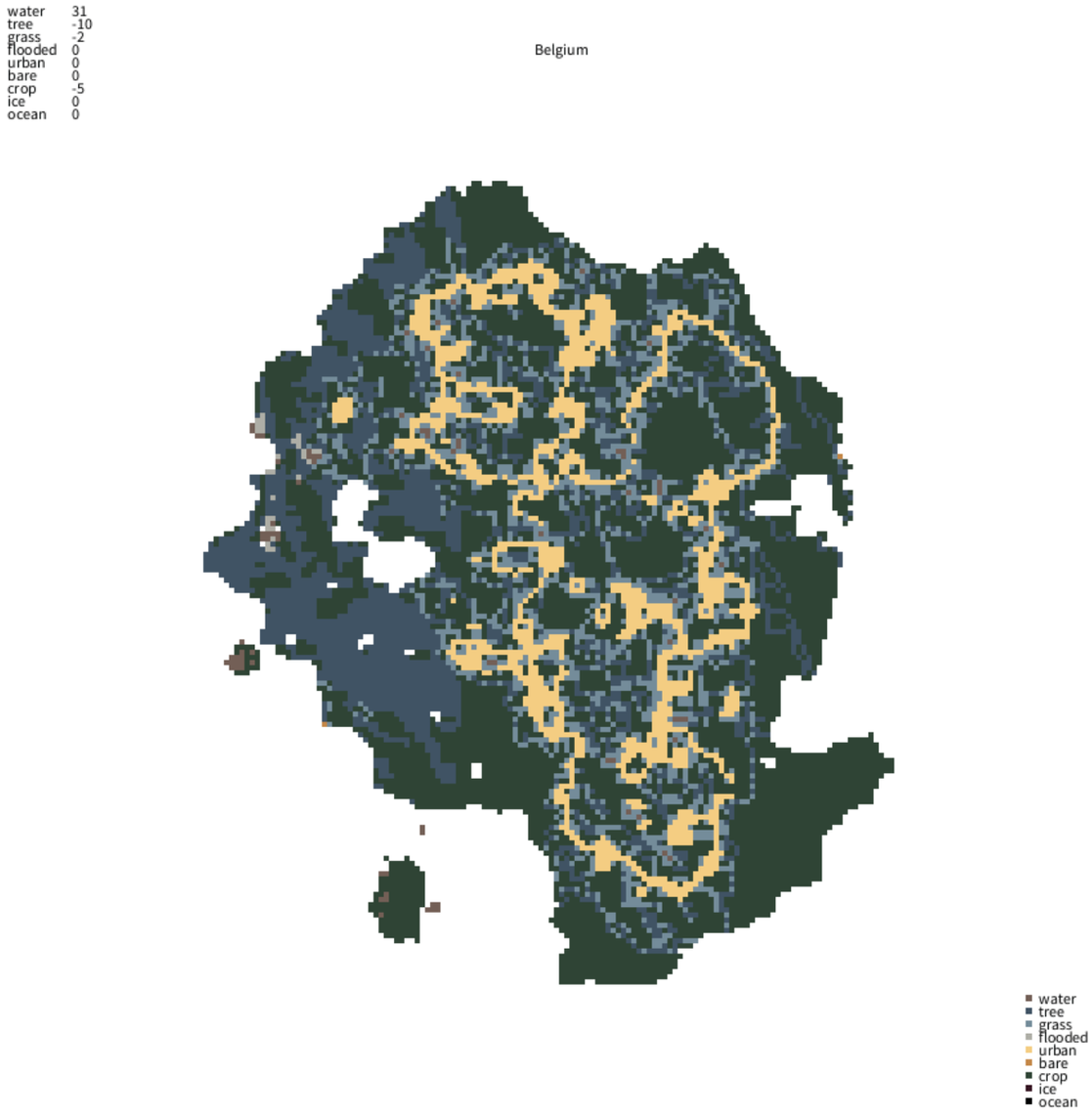
Building islands

After setting up the “wave function collapse” system, I started working on the islands. Generating islands in most game systems is done by using a simple perlin noise function, using the brightness as the height of the terrain (1). This would cause the terrain to go up to the edge of the display, and while this can be useful when creating larger maps, when creating islands we add an extra function,. This makes sure pixels are darkened as they get close to the edge of the screen (2). Multiplying these values leaves you with a perfect blueprint of an island. In order to make the island more rugged; decrease the noise scale, otherwise increase the scale to make the outline of the island more smoothly.

After the island is generated, it is divided into smaller squares to later apply WFC on. When I did the first experiments, I’ve noticed most generations were very illogical and did little to convince the audience of the tangibility of the island.

In order to solve this, I’ve looked at the actual world, and tried to implement some rules that would make the island appear more natural, more realistic.



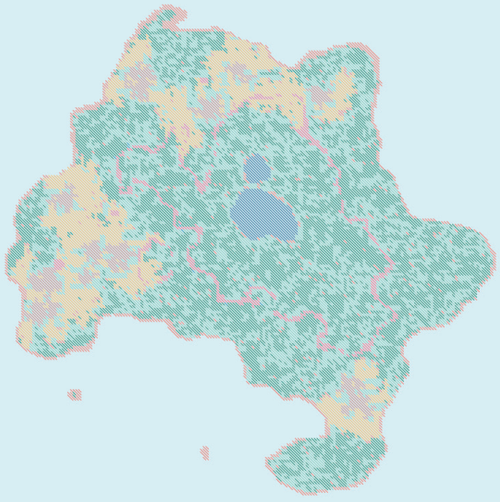


Realism

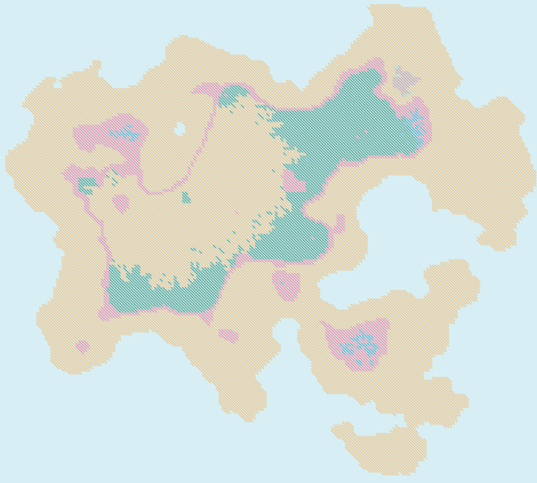
The spark came from the simple realization ice would seldom be close to the sea, unless all you have is ice. Ice normally would occur on higher altitudes. When we look at the noise map we generated in the previous step, the brightest colors would be the highest points, so everything above a certain point can be assigned to “ice”, based on the percentage coverage of the ice in the data. This also means the neighboring tiles (the ones just not high enough to be ice) collapse to their respective fields (in this case; bare, trees or grasslands). Next I’ve selected the lowest points and made the into the “water bodies” field, in accordance with the data. Just beside these water fields could be the flooded fields.

Next, I’ve tried systems to place the urban landscape. Logic dictates this would be close to water, So I tried placing them just above the water, but this resulted in these long stringing flows of urban space (which were quite accurate in Belgium, but not so much in other countries). Eventually, by tweaking the rules (encouraging it to place fields of similar kind next to each other) made it possible to place a few spots of “urban” environment, and letting the system place the other fields next or close to the first ones naturally.

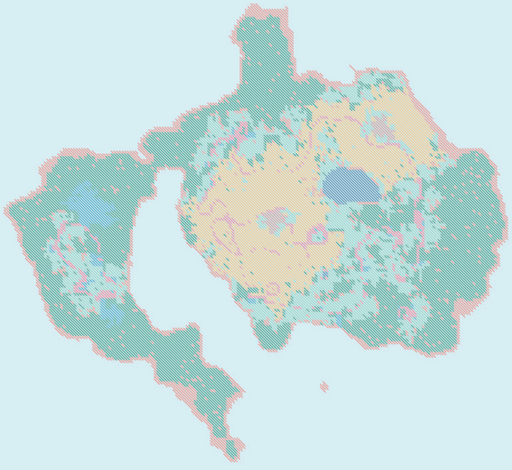
Switzerland



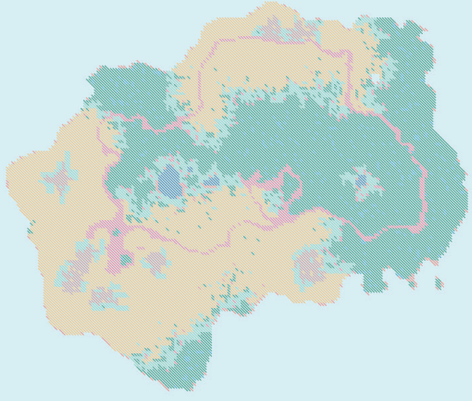
Timor-Leste



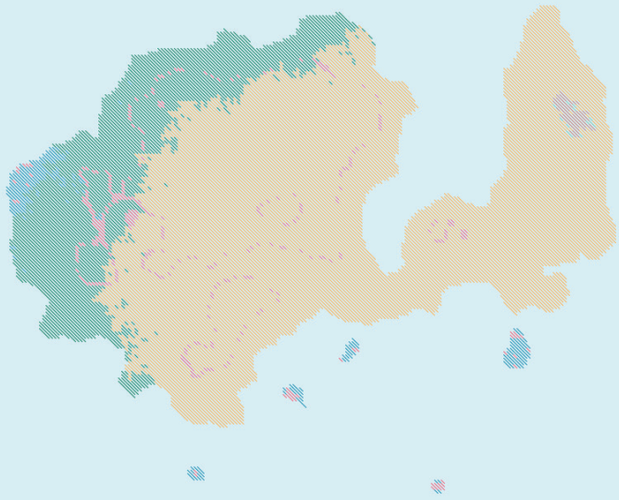
world.csv



Europe



Rwanda



Data insertion

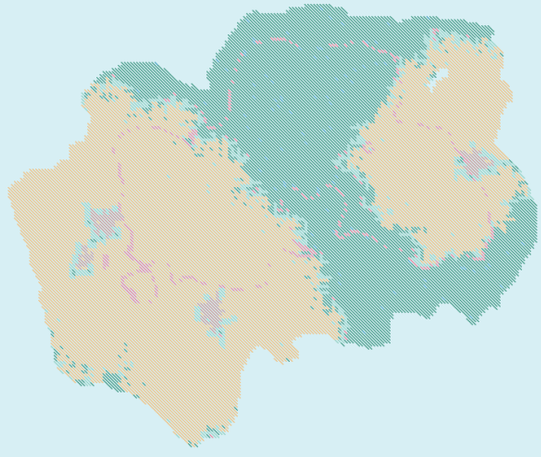
At this point, the islands needed to reflect the data of the country it represents. In order to do this, a few steps are taken;

At the start, the island is formed, and divided into squares (fields). The fields are then counted and kept in an array to track.

Based on the total amount of squares available for the island, a list is kept of the division of fields for this island. If for example 10% of the country surface consists of forests, 10% of the total amount of squares will be dedicated to this.

For every square placed, the list is updated (when a particular field has been used, the amount corresponding to it is diminished by one). When an item in this list has been reduced to 0, all tiles are updated to no longer include this option in their WFC decision making.

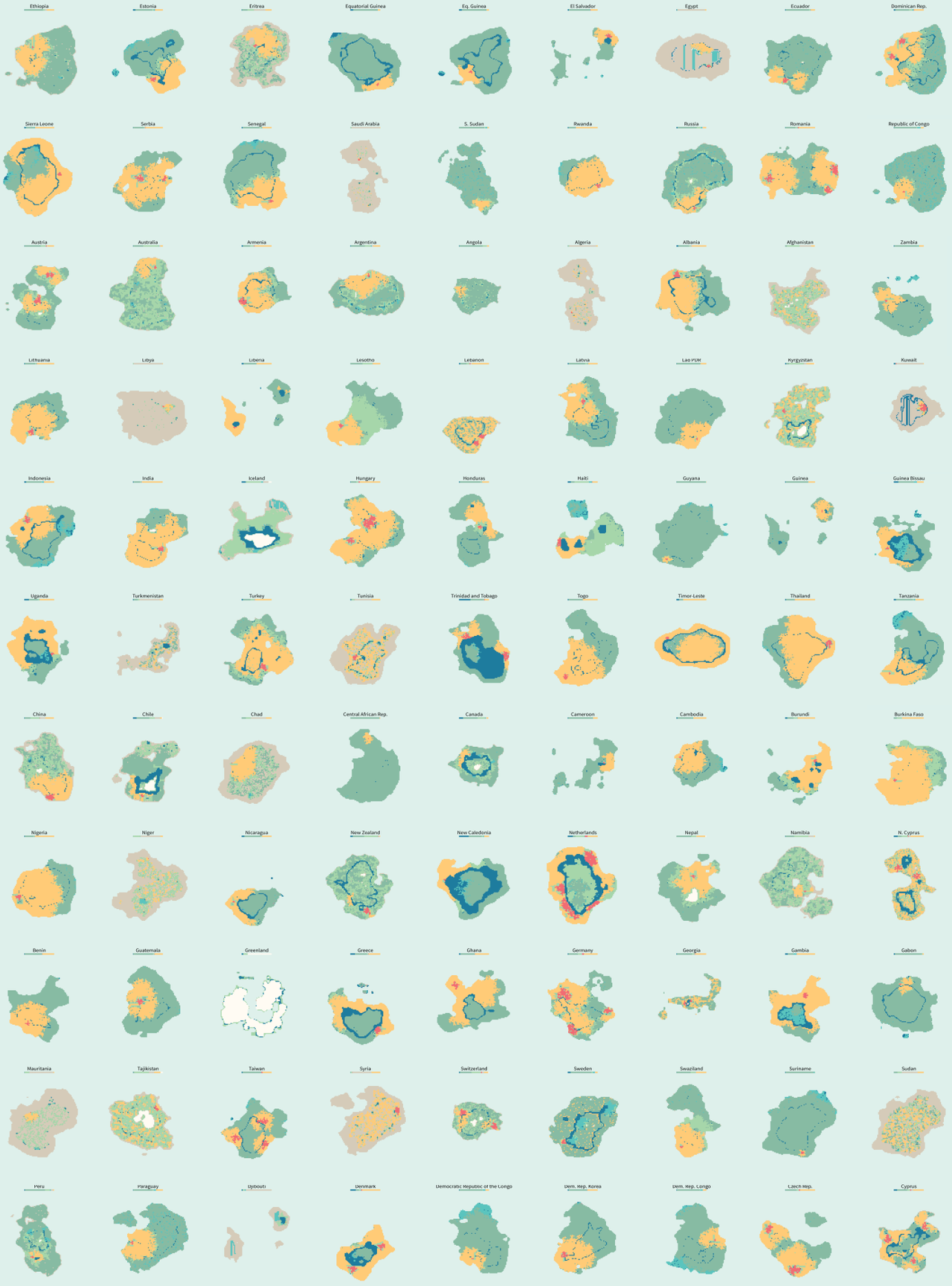
Lithuania

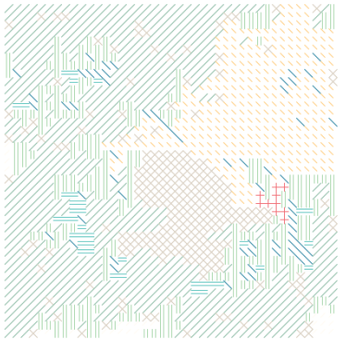


DEAGEA

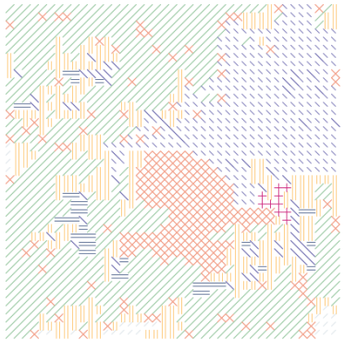
DIVIDED EARTH

Every country in the world, envisioned as an island of normalised size, with the same comparable land composition as the source country

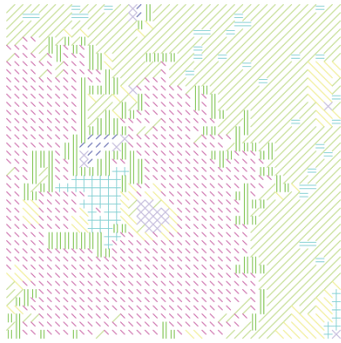

Ethiopia
Estonia
Eritrea
Equatorial Guinea
Eq. Guinea
El Salvador
Egypt
Ecuador
Dominican Rep.
Sierra Leone
Serbia
Senegal
Saudi Arabia
S. Sudan
Rwanda
Russia
Romania
Republic of Congo
Austria
Australia
Armenia
Argentina
Angola
Algeria
Albania
Afghanistan
Zambia
Lithuania
Libya
Liberia
Lesotho
Lebanon
Latvia
Lao PDR
Kyrgyzstan
Kuwait
Indonesia
India
Iceland
Hungary
Honduras
Haiti
Guyana
Guinea
Guinea Bissau
Uganda
Turkmenistan
Turkey
Tunisia
Trinidad and Tobago
Togo
Timor-Leste
Thailand
Tanzania
China
Chile
Chad
Central African Rep.
Canada
Cameroon
Cambodia
Burundi
Burkina Faso
Nigeria
Niger
Nicaragua
New Zealand
New Caledonia
Netherlands
Nepal
Namibia
N. Cyprus
Benin
Guatemala
Greenland
Greece
Ghana
Germany
Georgia
Gambia
Gabon
Mauritania
Tajikistan
Taiwan
Syria
Switzerland
Sweden
Swaziland
Suriname
Sudan
Peru
Paraguay
Djibouti
Denmark
Democratic Republic of the Congo
Dem. Rep. Korea
Dem. Rep. Congo
Czech Rep.
Cyprus



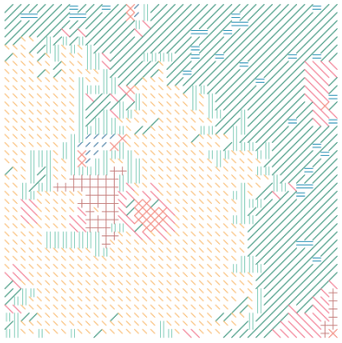
world



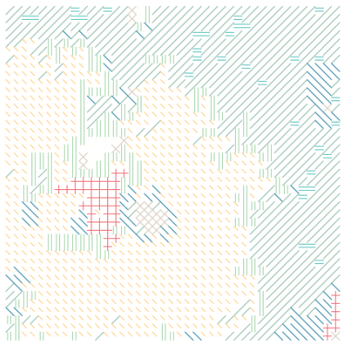
world



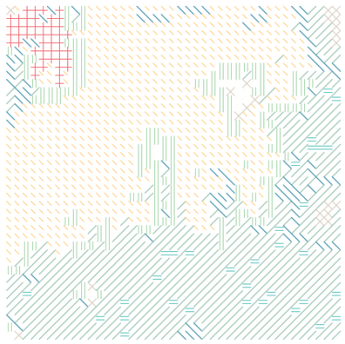
Europe



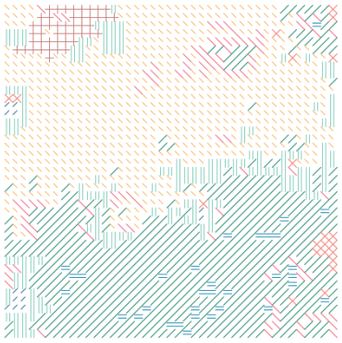
Europe



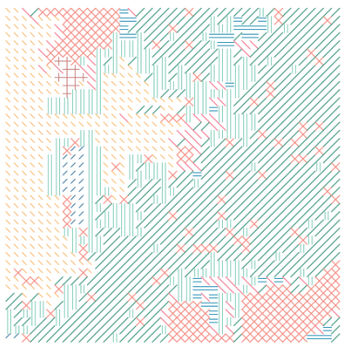
Europe



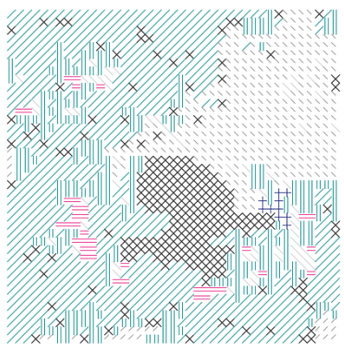
Europe



Europe



world



world

In these variations, I experimented with a stylized representation of the tiles, working more on texture than on the contents. This ended up being less readable, but could pave the way for relief impression or print on a generated map, to make the result more tangible.

Tile

An early version of the “tile” class, with an example of how “wave function collapse” could be implemented.

```
class Tile {
  int x, y, s, id;
  boolean active = true;
  ArrayList<Option> options = new ArrayList<Option>();
  Option picked = null;
  float elevation = -1;

  Tile(int i, int xp, int yp, int size, ArrayList<Option> o) {
    x = xp;
    y = yp;
    s = size;
    id = i;
    for (Option to : o) {
      options.add(to);
    }
  }
  void update() {
    for (Option co : country.options) {
      for (int i = 0; i < options.size(); i++) {
        if (options.get(i).handle == co.handle && co.left <= 0) {
          options.remove(i);
        }
      }
    }
  }
  void draw() {...}

  void pick() {
    if (options.size() > 0) {
      try {
        Tile[] tt = {
          tiles.get( ((x-1) * dim) + (y) ),
          tiles.get( ((x+1) * dim) + (y) ),
          tiles.get( (x * dim) + (y-1) ),
          tiles.get( (x * dim) + (y+1) )
        };
        Option highestChance = null;
        float h = -1;
        for (Option o : options) {
          o.chance = 0;
          for (Tile t : tt) {
            if (t!= null && t.picked!=null) {
              if (o.handle == t.picked.handle) {
                o.chance+= o.prio;
              }
            }
          }
          if (o.chance > h) {
            h = o.chance;
            highestChance = o;
          }
        }
        if (h >=2) {
          picked = highestChance;
        } else {

          picked = options.get(floor(random(options.size())));
        }
        options = new ArrayList<Option>();
        country.subtract(picked);

        for (Tile t : tt) {
          if (t!= null) {
            t.removeOnOption(picked);
          }
        }
      }
      catch(IndexOutOfBoundsException e) {
        println(e);
      }
    }
  }

  void pick(Option p) {
    picked = p;
    options = new ArrayList<Option>();
    country.subtract(p);

    // up
    if (y != 0) {
      tiles.get( (x * dim) + (y-1) ).removeOnOption(picked);
    }
    if (y != dim-1) {
      tiles.get( (x * dim) + (y+1) ).removeOnOption(picked);
    }

    if (x != 0) {
      tiles.get( ((x-1) * dim) + (y) ).removeOnOption(picked);
    }
    if (x != dim-1) {
      tiles.get( ((x+1) * dim) + (y) ).removeOnOption(picked);
    }
  }


  void removeOnOption(Option p) {
    int checking = 0;
    while (checking != options.size()) {
      boolean found = false;
      for (int o : p.allows) {

        if (o == options.get(checking).id) {
          found = true;
        }
      }
      if (!found && options.get(checking).id != p.id) {
        options.remove(checking);

        if (options.size() == 1) {
          pick(options.get(0));
        }
      } else {
        checking++;
      }
    }
  }
}
```

ESA outcome

On the right is the final poster, as it was submitted to the ESA competition. On 1 December, I received note that I ended up in the final 10 selections, and it has been posted in the gallery on the ESA little pictures gallery (https://climate.esa.int/en/little-pictures-gallery/).

The email I've received:

Good morning Jan,

Your Little Picture was very highly commended by our judges and made it into the top 10 entries to ESA's Little Picture of Climate competition.

This is a real achievement as we received well over 400 submissions from across Europe.

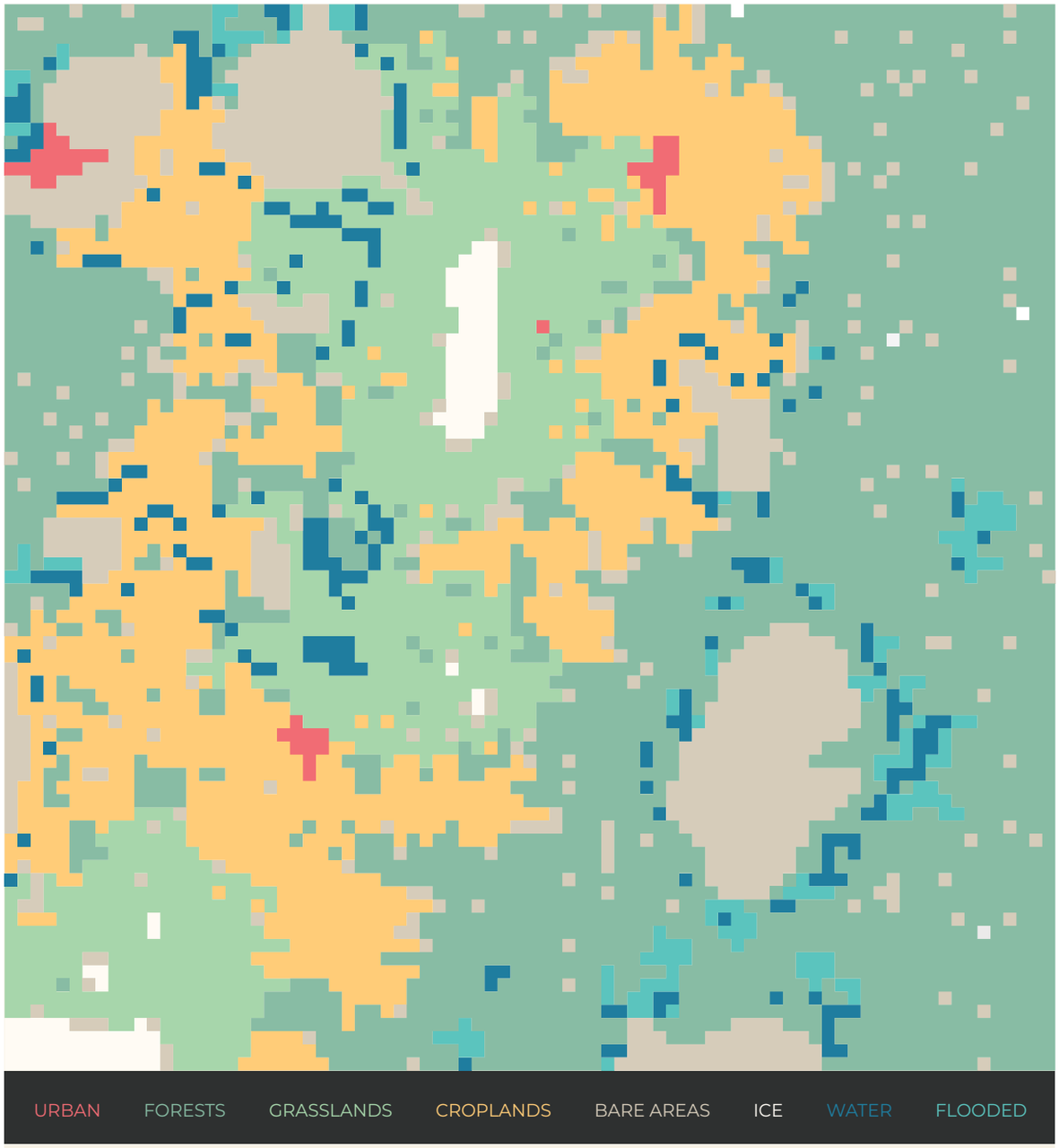
Next steps:

ESA will be showing this on the ESA booth at COP28 and will plan a news item to be published on the 9 Dec to coincide with the COP28 land use daily theme. We may include your entry depending on editorial decisions we are making right now and will add your Little Picture to our gallery (https://climate.esa.int/en/little-pictures-gallery/) where we will add your little picture (with the judges commendations).

Our social media team will be in contact with you directly potentially and will circulate your little picture online.

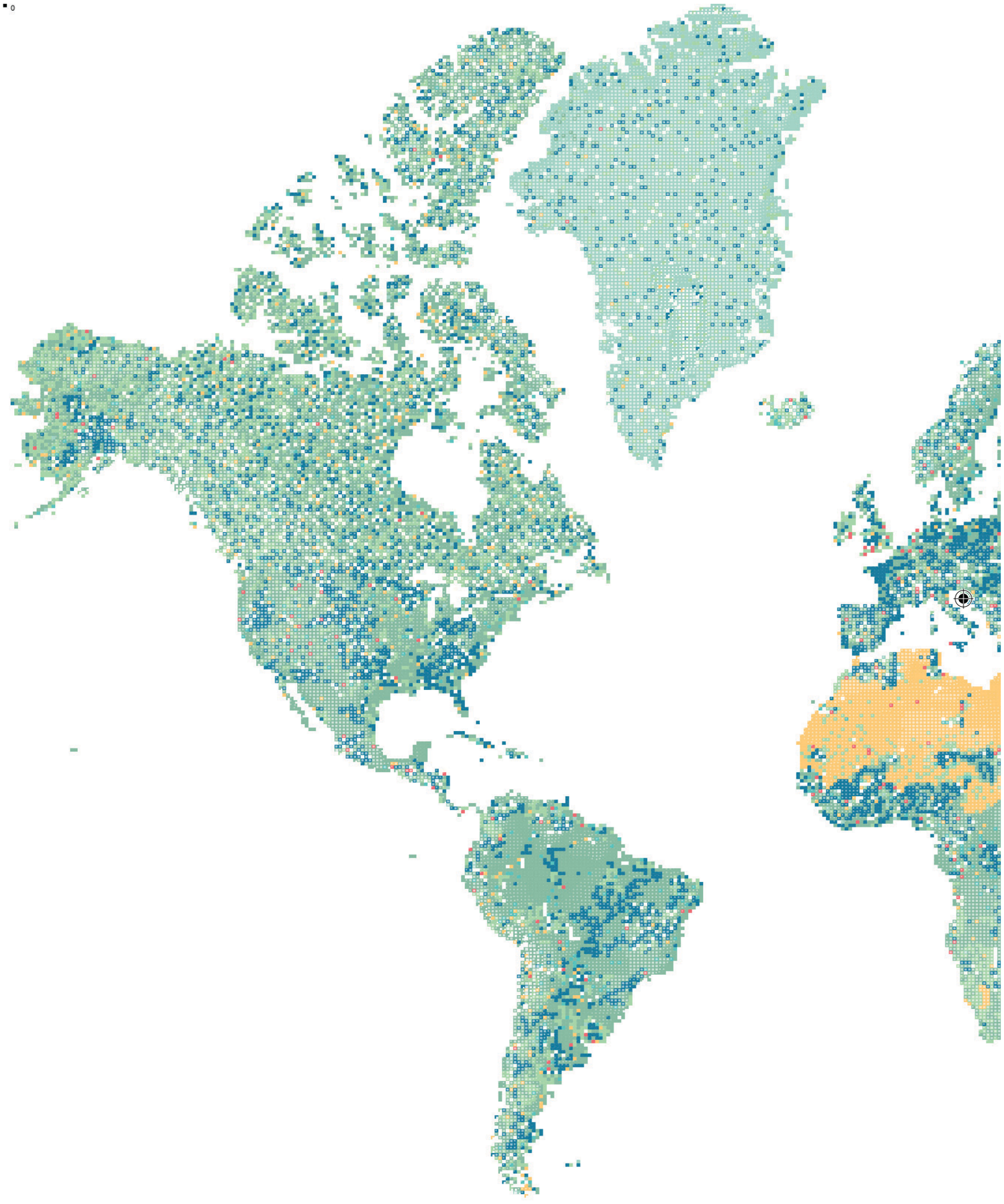
Thanks again for taking part, we were thrilled to receive such well thought through entries from you all.

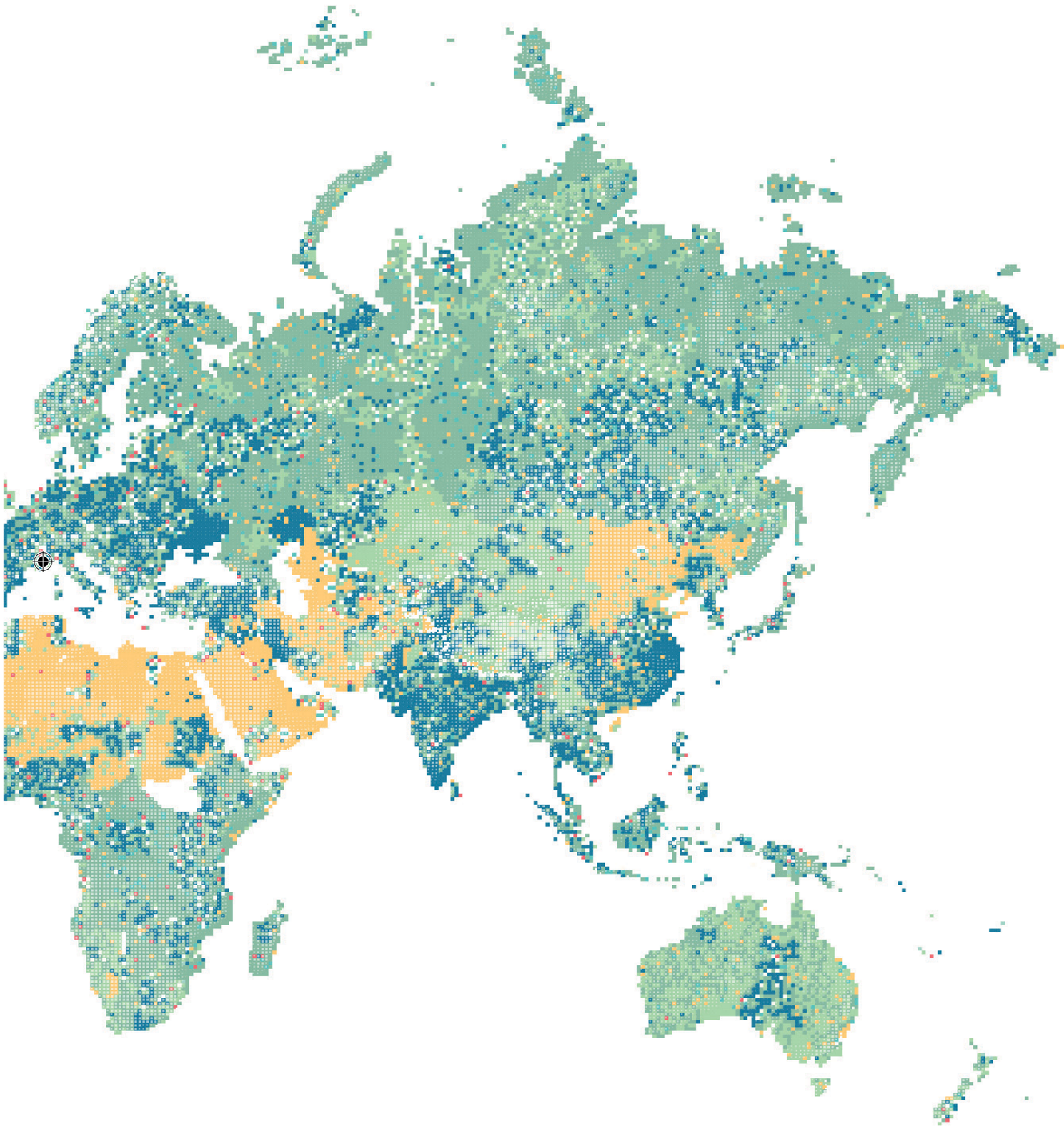
Regards



How we use our land

To illustrate the extent of our footprint on the land, this visualization depicts the allocation of space for different purposes. Urban areas are represented in red, while the yellow zones indicate crop fields required to sustain the urban population. The map highlights the disproportionate ratio between the land utilized for crop cultivation and our actual living spaces.





Coding process

During the entire process, about 56 processing sketches, all different iterations around the same dataset. These sketches include the various experiments, the iterations on the "Bring your own beamer" project, and the earliest depictions of the data.

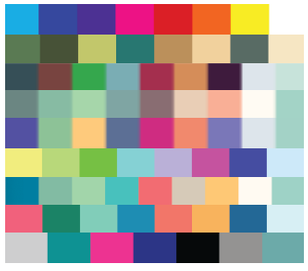
At the final stages, I updated and streamlined the process to make iteration quicker. At this point, I was generating world maps where earlier I was just working with countries. The main change to the code was to assign tiles to a country, so I could look up the amount of tiles left on the fly, but this slowed the generation time significantly. In order to make it possible to experiment with the colors and patterns without having to regenerate the maps, I split the sketch in two parts. One part takes care of the assignment to countries (above), and the wave function collapse. The other sketch iterates over the individual pixes, reading out the data, and scaling it up to include the patterns.

The first sketch starts with a grid of voxels, defined by the resolution. Using a SVG of a world map (named) it iterates over each pixel and assigns the ID of the country to the blue channel. This also prepares the wave function collapse, as instead of islands, we have arrays of tiles belonging to a country. The WFC algorithm runs until all the tiles have an assignment (the tiles not belonging to a country are removed), and the type of tile is stored in the Red channel of the pixel, Finally, the Green channel is used to store the height map of the pixel at that location.

This exports a map as seen on the right, which can be used as the blueprint for the next sketch. This sketch will enlarge each pixel by a factor of 14, in order to be able to display the patterns. It will use an image as the color palette, and us the data stored in the pixels to generate a final map, which it can export as svg, png, tiff, etc.

This process allows for quick iteration on for example the color palettes (bellow), the patterns or the placements.

Eventually, the first sketch also became responsible to calculate whether the data of the type displayed has gone up or down, and the green value shifted to include this as well. The second sketch was adjusted to show this instead of altitude.



File names (Latest first)

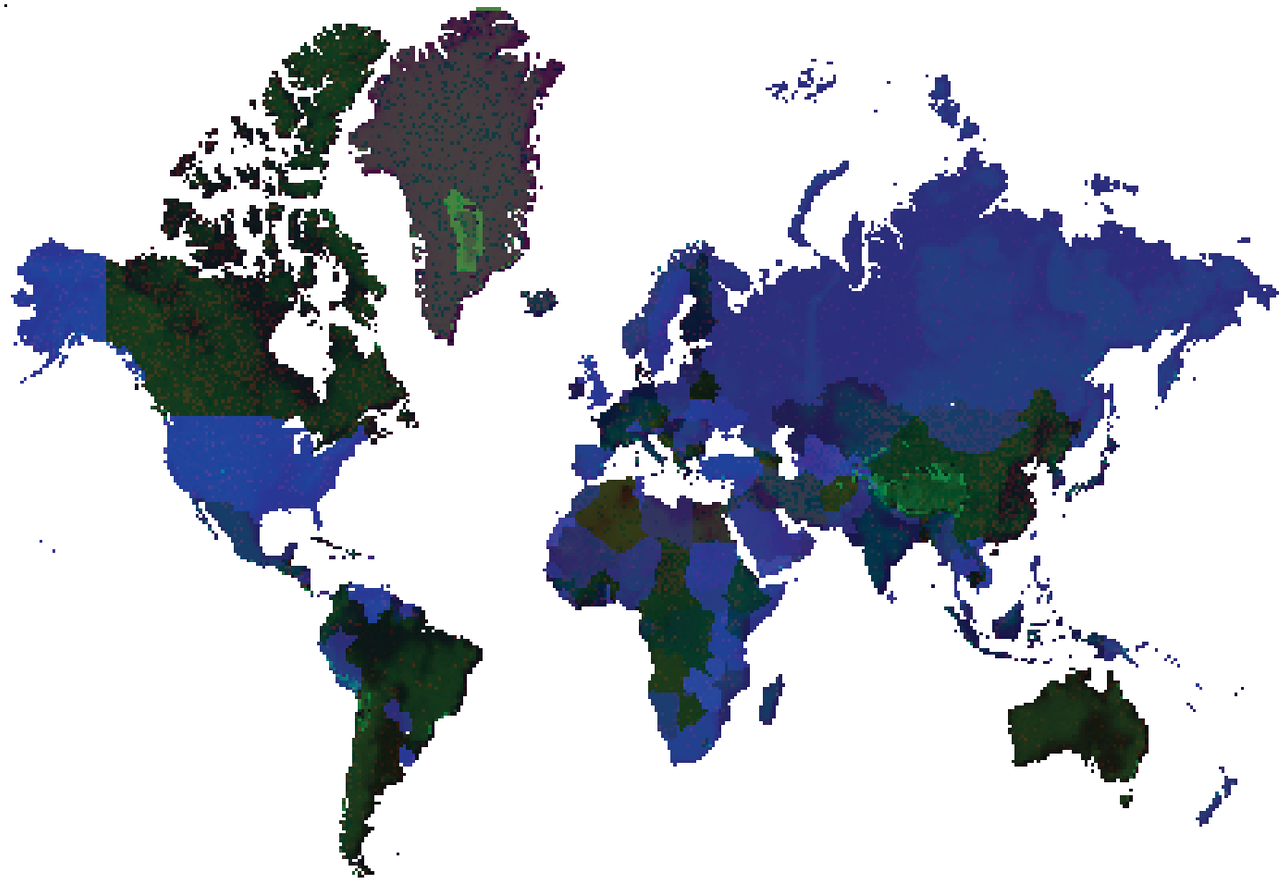
(sorry about the lack of naming convention)

dataInserter
tilemaker
WFC_Country_3
WFC_Countrylabels
WFC_Country_2
old
uitestW
WFC_Country
WFC_Country_random
LittlepicturesWFC
heightMapTest
LUCA_A1_physicalMapExplorer
simplepictures_poster_landplacement_print
walker_images
LUCA_A2_V2

WFC_v40
WFC_v39
WFC_v38
WFC_v37_cubed
LUCA_A2_V1
WFC_AE_1
WFC_v34_cubed
boids_insta
boids_header
simplepictures_poster_landplacement_beurs
WFC_v36
WFC_world
WFC_v35_auto
WFC_v34
WFC_v33

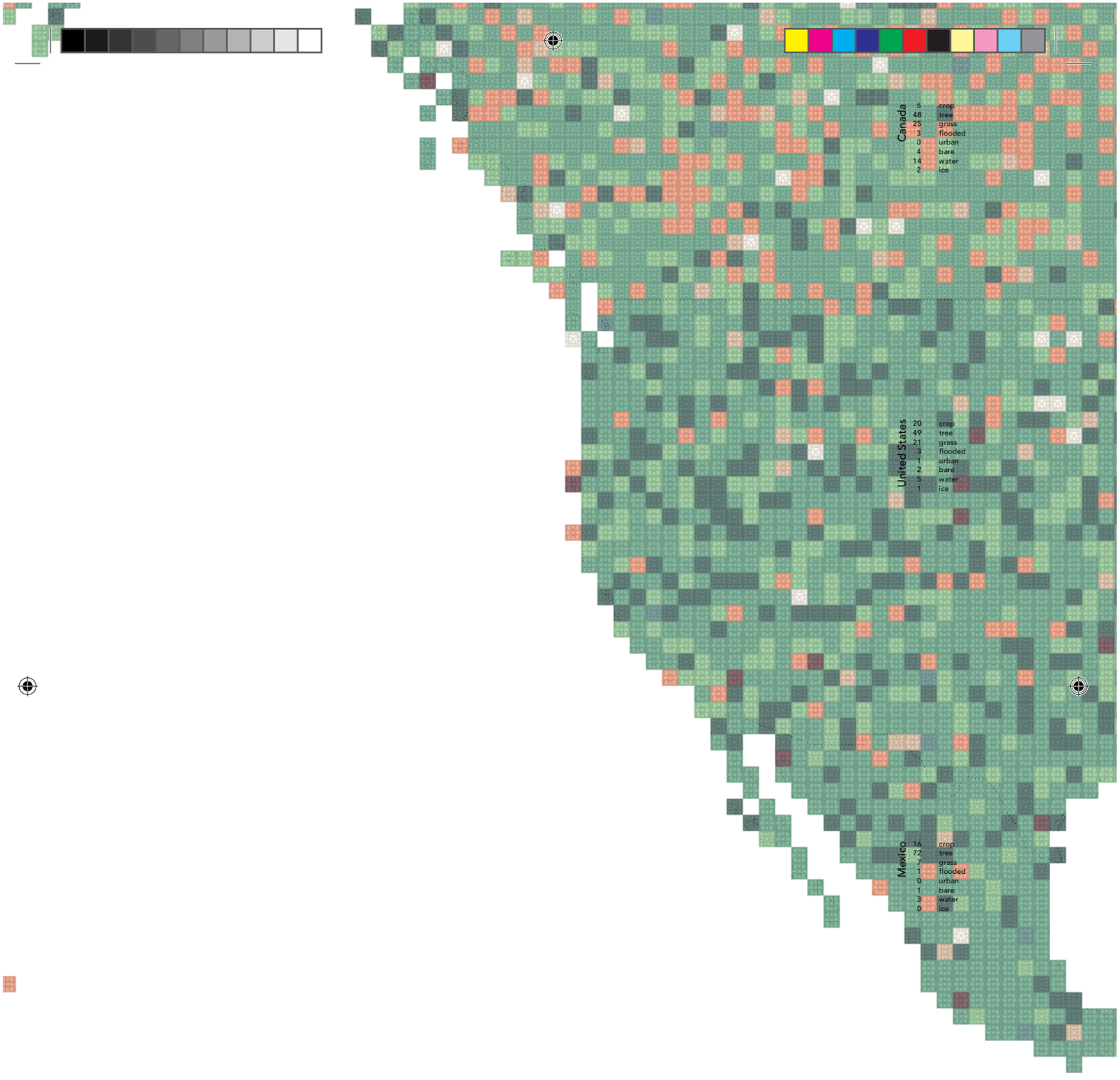
WFC_v32
WFC_v3
WFC_v2
WFC_v1
country_data_midi_gen4_spread
boids_poster
posterCheck_atelier
country_data_midi_gen4
country_data_midi_gen3
country_data_midi_gen2
simplepictures_poster_landplacement
country_data_midi_gen1
country_data_midi_play_gr
country_data_midi_play_random3D
country_data_midi_play_3e

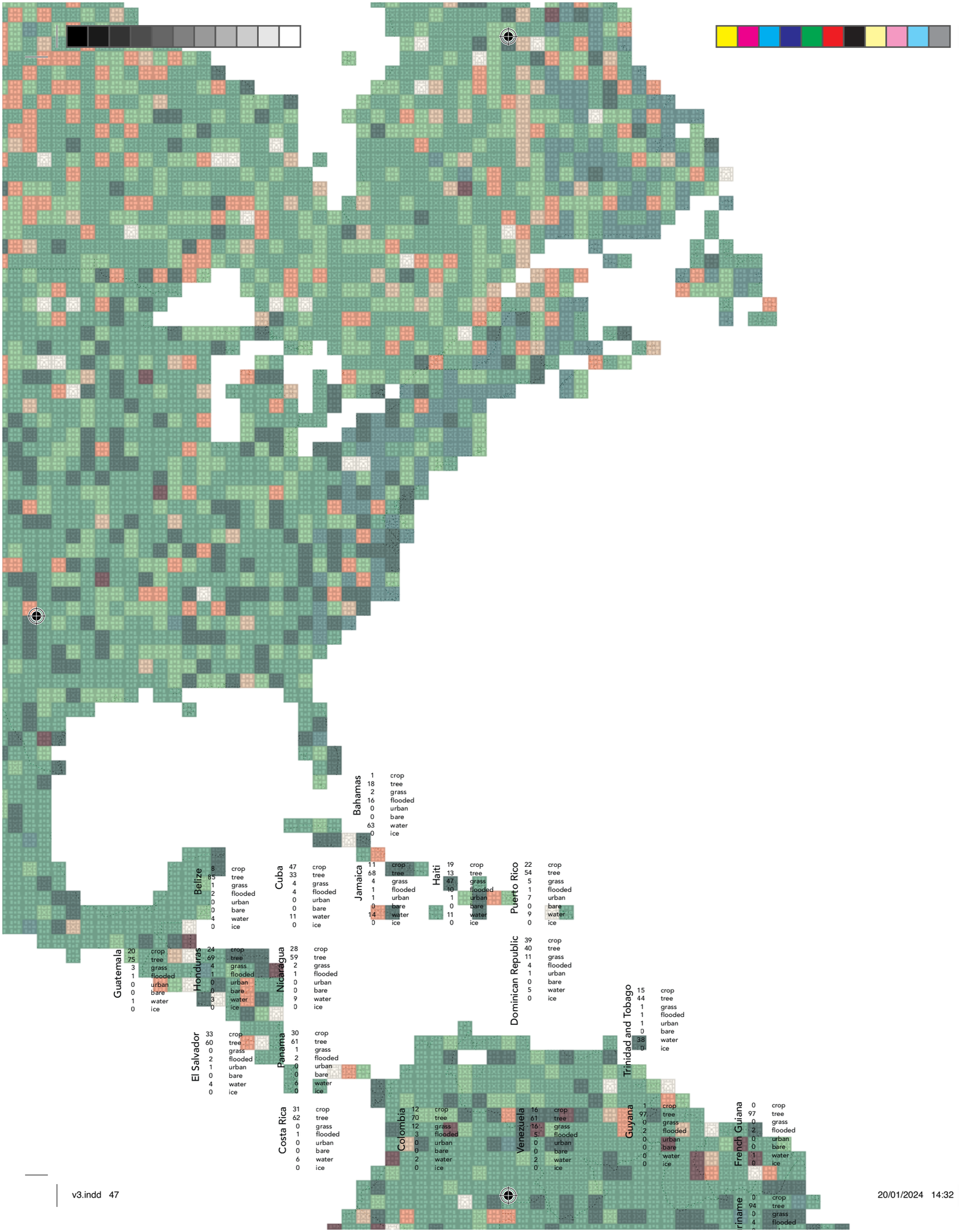
country_data_midi_play
country_data_midi
simplepictures_poster_aggr_bars_print
simplepictures_poster_aggr_perc
simplepictures_poster_aggr_1
simplepictures_poster_aggr_2
simplepictures_poster_aggr_bars
simplepictures_poster_aggr
simplepictures_poster
simplepictures_exp_aggr2
simplepictures_exp_aggr
simplepictures_exp
palletegen



End result

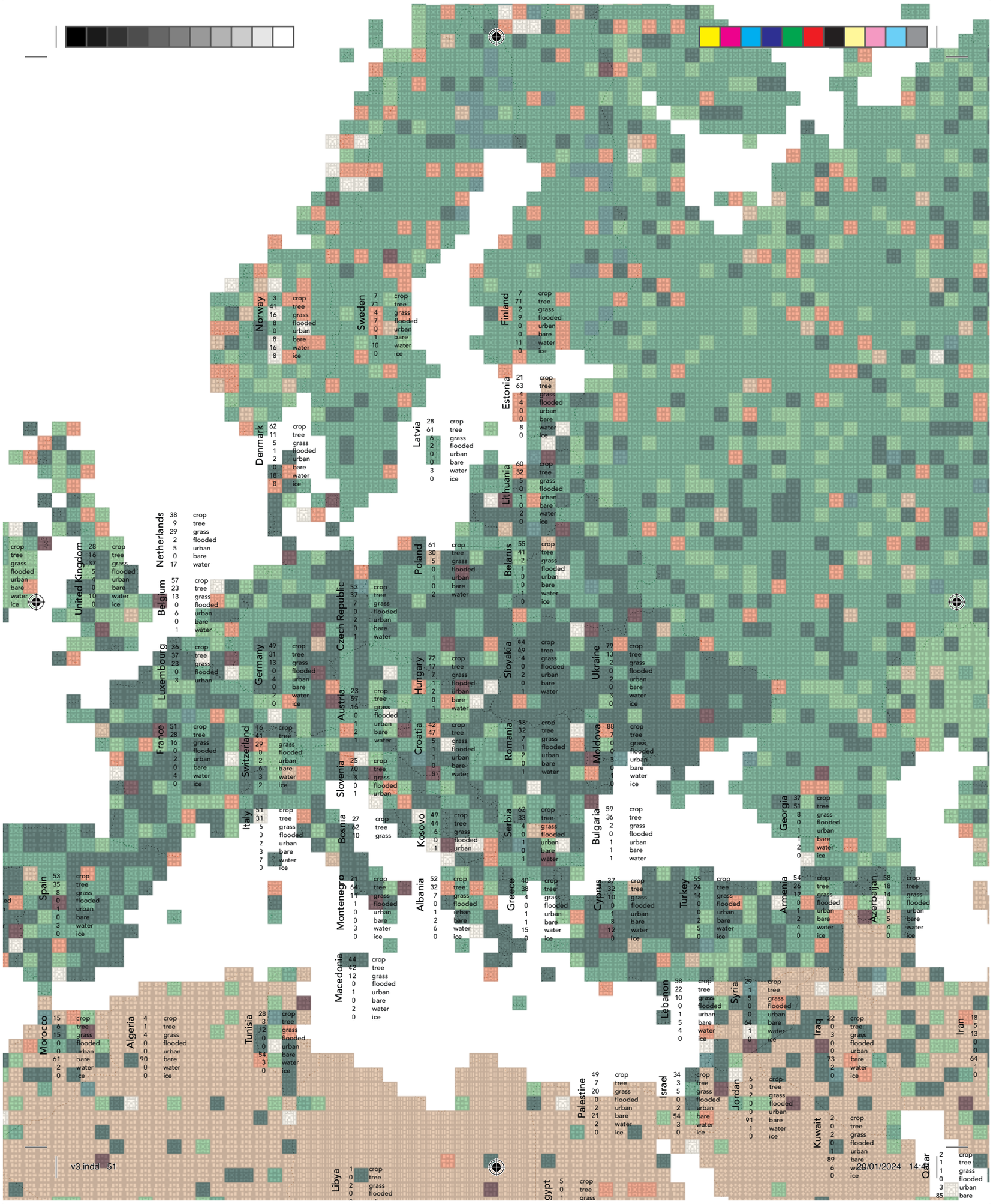
The next pages show the end result. The black data were printed in UV reflective neon yellow, and the scale in real life is smaller to keep within the limits of the magnification tool. On the back cover, a small legend is applied in the upper right corner, this legend can be used in combination with the maps on the following pages.

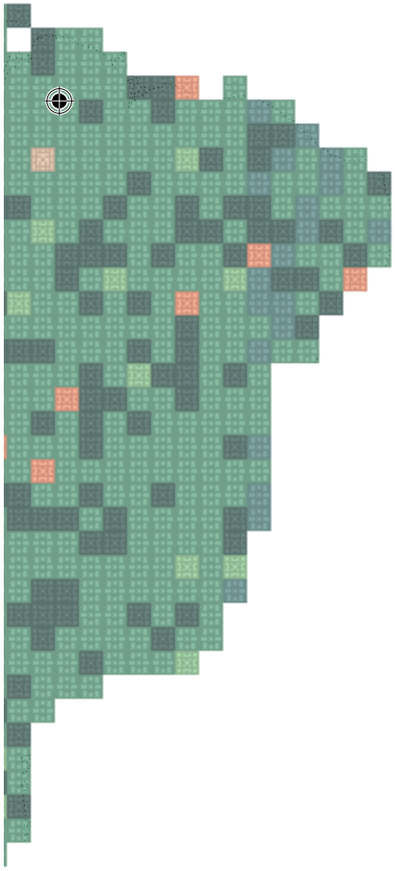

Canada
6 crop
48 tree
25 grass
3 flooded
0 urban
4 bare
14 water
2 ice
United States
20 crop
49 tree
21 grass
3 flooded
1 urban
2 bare
5 water
1 ice
Mexico
16 crop
72 tree
7 grass
1 flooded
0 urban
1 bare
3 water
0 ice

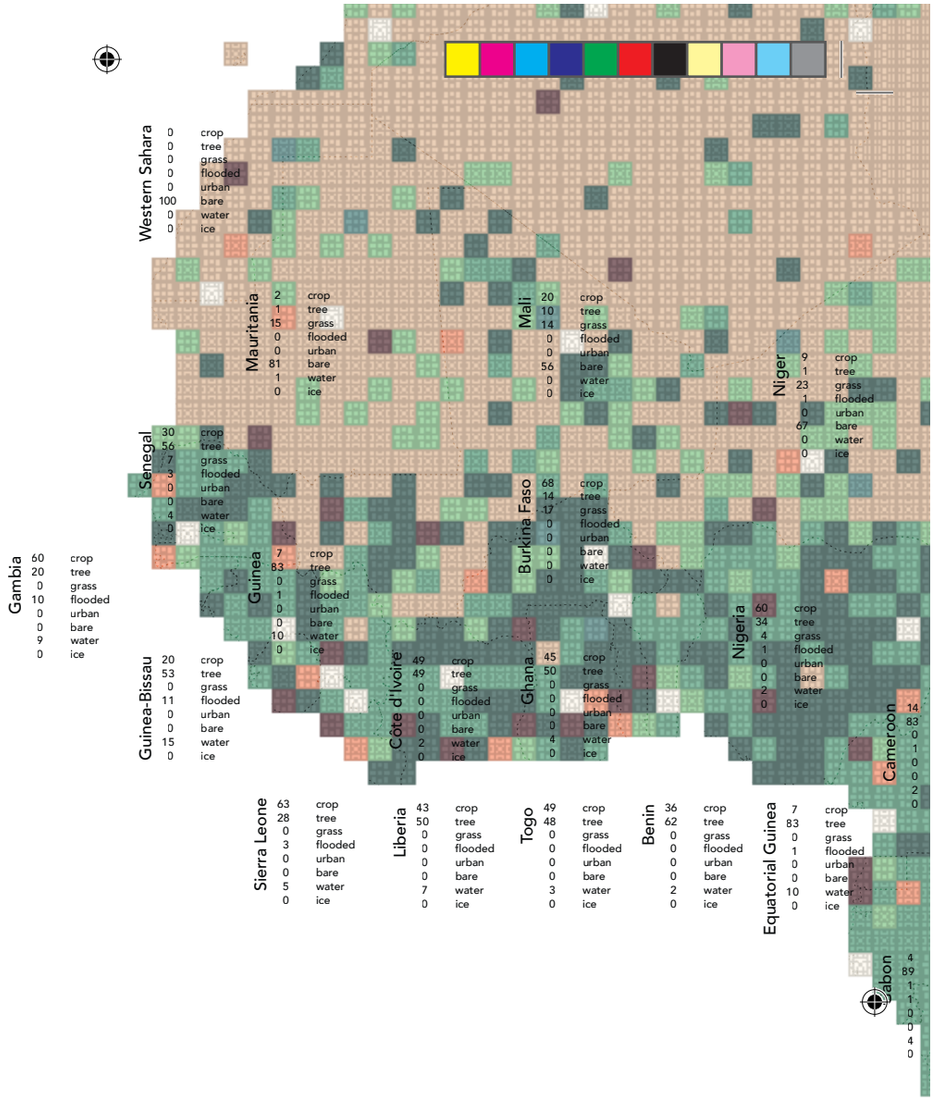

Bahamas
1 crop
18 tree
2 grass
16 flooded
0 urban
0 bare
63 water
0 ice
Belize
8 crop
85 tree
1 grass
2 flooded
0 urban
0 bare
4 water
0 ice
Cuba
47 crop
33 tree
4 grass
4 flooded
0 urban
0 bare
11 water
0 ice
Jamaica
11 crop
68 tree
4 grass
1 flooded
1 urban
0 bare
14 water
0 ice
Haiti
19 crop
13 tree
47 grass
10 flooded
1 urban
0 bare
11 water
0 ice
Puerto Rico
22 crop
54 tree
5 grass
1 flooded
7 urban
0 bare
9 water
0 ice
Guatemala
20 crop
75 tree
3 grass
1 flooded
0 urban
0 bare
1 water
0 ice
Honduras
24 crop
69 tree
4 grass
1 flooded
0 urban
0 bare
3 water
0 ice
Nicaragua
28 crop
59 tree
2 grass
1 flooded
0 urban
0 bare
9 water
0 ice
Dominican Republic
39 crop
40 tree
11 grass
4 flooded
1 urban
0 bare
5 water
0 ice
El Salvador
33 crop
60 tree
0 grass
2 flooded
1 urban
0 bare
4 water
0 ice
Panama
30 crop
61 tree
1 grass
2 flooded
0 urban
0 bare
6 water
0 ice
Trinidad and Tobago
15 crop
44 tree
1 grass
1 flooded
1 urban
0 bare
38 water
0 ice
Costa Rica
31 crop
62 tree
0 grass
1 flooded
0 urban
0 bare
6 water
0 ice
Colombia
12 crop
70 tree
12 grass
3 flooded
0 urban
0 bare
2 water
0 ice
Venezuela
16 crop
61 tree
16 grass
5 flooded
0 urban
0 bare
2 water
0 ice
Guyana
1 crop
97 tree
0 grass
2 flooded
0 urban
0 bare
0 water
0 ice
French Guiana
0 crop
97 tree
0 grass
2 flooded
0 urban
0 bare
1 water
0 ice
Suriname
0 crop
94 tree
0 grass
4 flooded

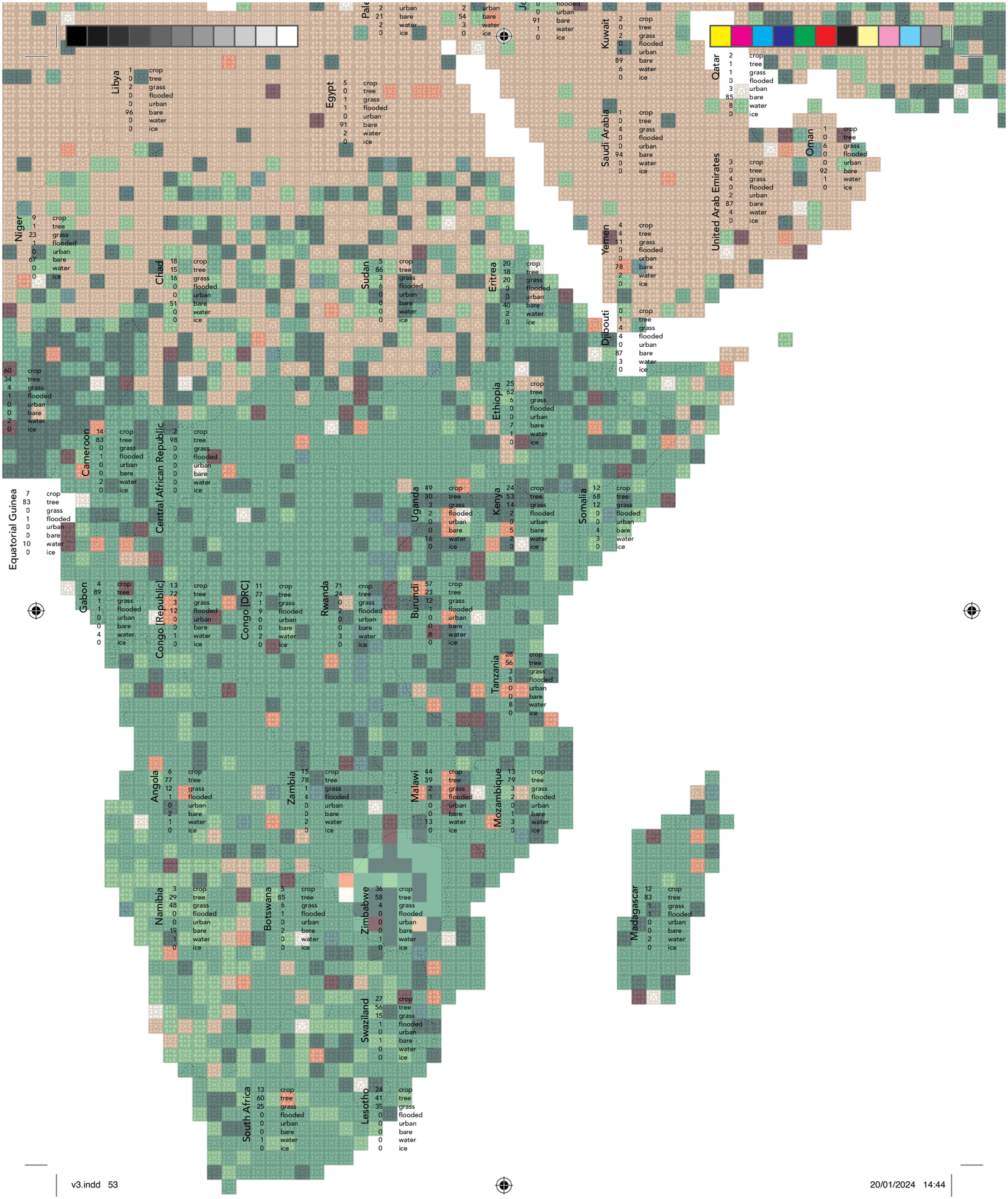
Bahamas
Jamaica
Cuba
Haiti
Puerto Rico
Dominican Republic
Belize
Honduras
Nicaragua
Guatemala
El Salvador
Panama
Costa Rica
Colombia
Venezuela
Guyana
Trinidad and Tobago
French Guiana
Suriname
Ecuador
Peru
Bolivia
Brazil
crop
tree
grass
flooded
urban
bare
water
ice

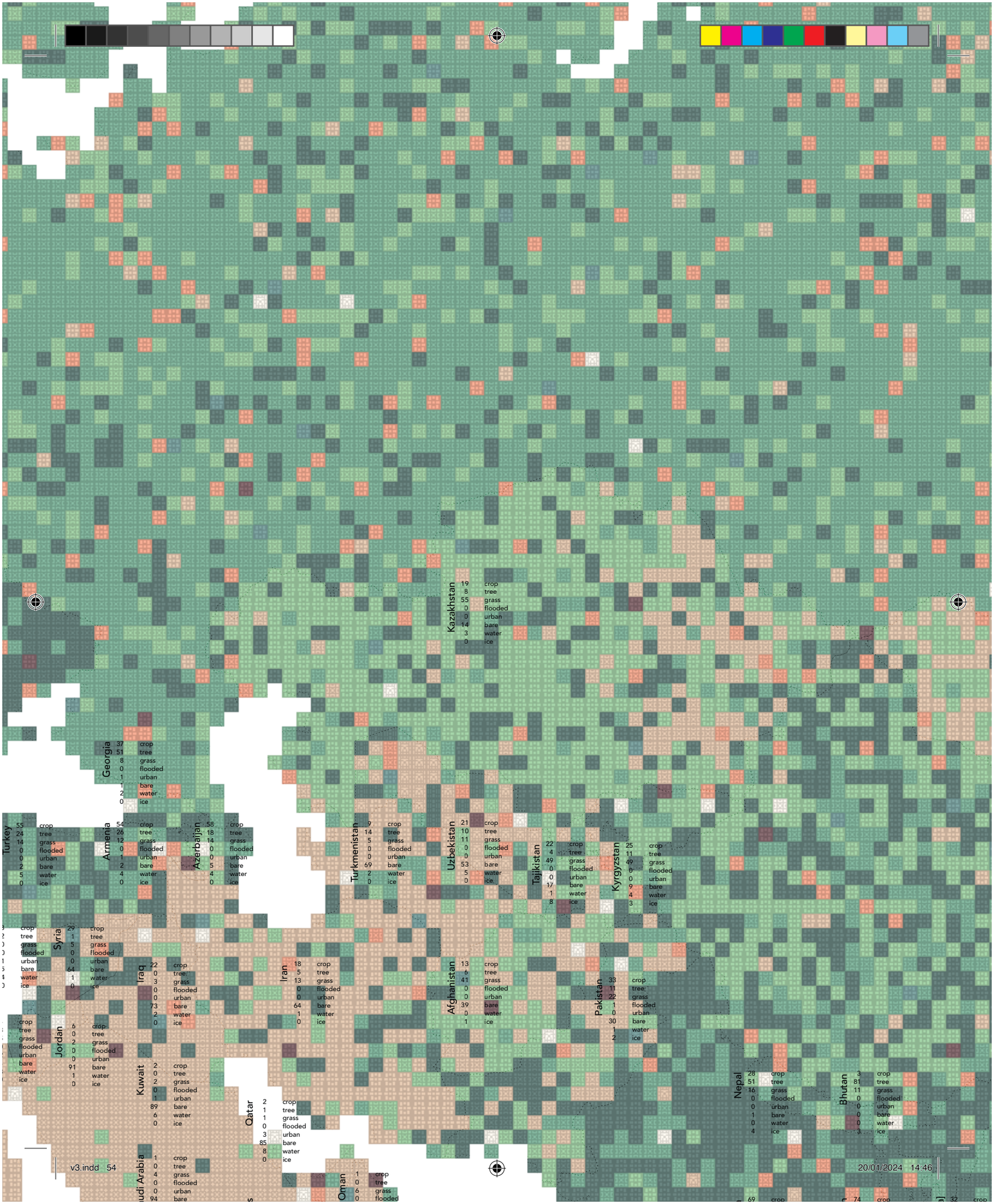
Chile
crop 6
tree 43
grass 18
flooded 0
urban 0
bare 17
water 12
ice 4
Argentina
crop 22
tree 49
grass 20
flooded 3
urban 0
bare 4
water 2
ice 0
Uruguay
crop 22
tree 112
grass 63
flooded 2
urban 0
bare 0
water 2
ice 0
Brazil
crop 27
tree 49
grass 5
flooded 3
urban 0
bare 0
water 2
ice 0
Paraguay
crop 18
tree 71
grass 3
flooded 7
urban 0
bare 0
water 1
ice 0
Bolivia
crop 5
tree 69
grass 14
flooded 3
urban 0
bare 8
water 1
ice 0
Islas Malvinas
crop 0
tree 5
grass 56
flooded 0
urban 0
bare 0
water 38
ice 0

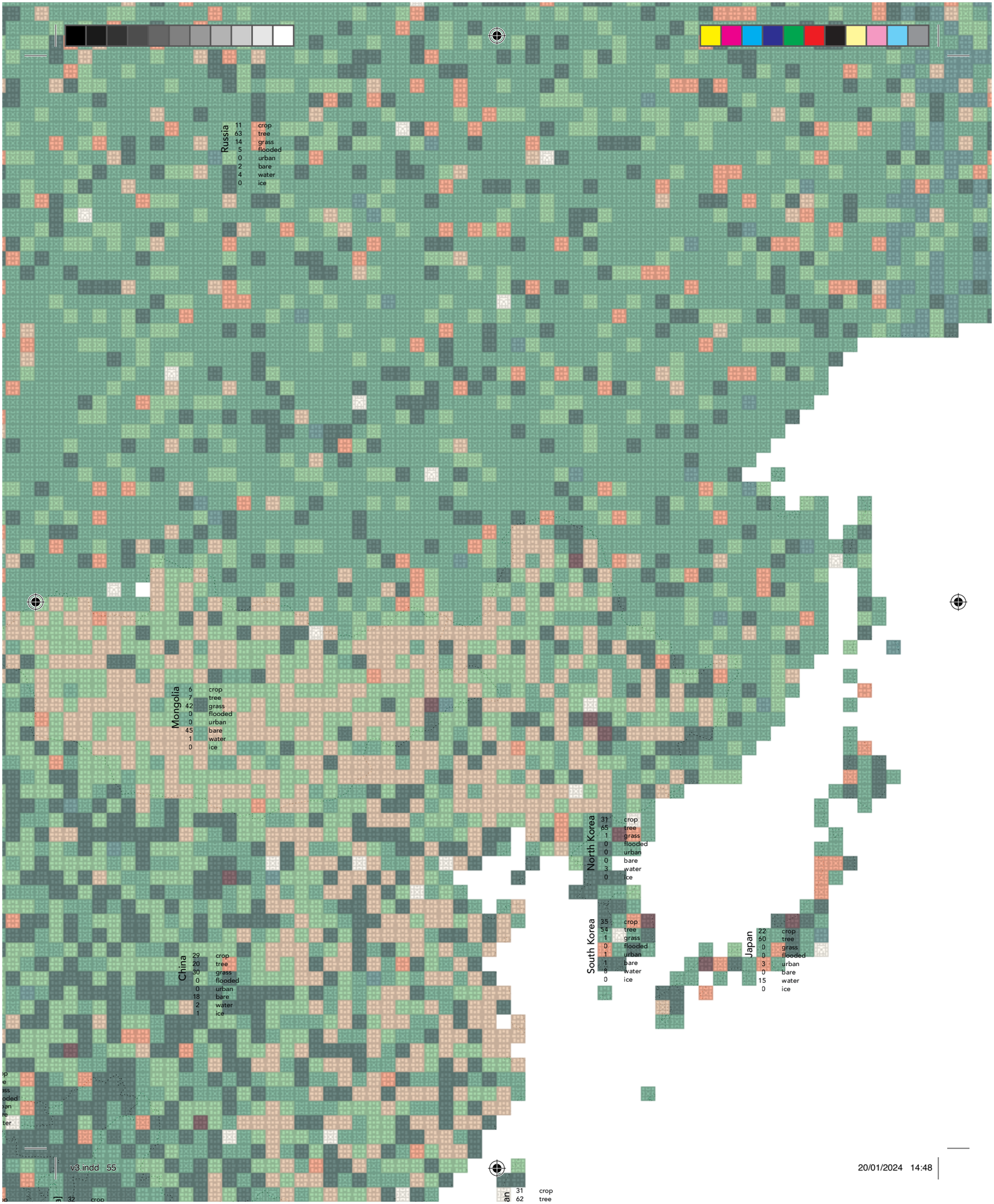

Norway
3 crop
41 tree
16 grass
8 flooded
0 urban
8 bare
16 water
8 ice
Sweden
7 crop
71 tree
4 grass
7 flooded
0 urban
1 bare
10 water
0 ice
Finland
7 crop
71 tree
2 grass
9 flooded
0 urban
0 bare
11 water
0 ice
Estonia
21 crop
63 tree
4 grass
4 flooded
0 urban
0 bare
8 water
0 ice
Latvia
28 crop
61 tree
6 grass
2 flooded
0 urban
0 bare
3 water
0 ice
Lithuania
60 crop
32 tree
5 grass
0 flooded
1 urban
0 bare
2 water
0 ice
Denmark
62 crop
11 tree
5 grass
1 flooded
2 urban
0 bare
18 water
0 ice
Netherlands
38 crop
9 tree
29 grass
2 flooded
5 urban
0 bare
17 water
United Kingdom
28 crop
16 tree
37 grass
5 flooded
4 urban
0 bare
10 water
0 ice
Belgium
57 crop
23 tree
13 grass
0 flooded
6 urban
0 bare
1 water
Poland
61 crop
30 tree
5 grass
0 flooded
1 urban
0 bare
2 water
Belarus
55 crop
41 tree
2 grass
1 flooded
0 urban
0 bare
1 water
0 ice
Czech Republic
53 crop
37 tree
7 grass
0 flooded
2 urban
0 bare
1 water
Luxembourg
36 crop
37 tree
23 grass
0 flooded
3 urban
Germany
49 crop
31 tree
13 grass
0 flooded
4 urban
0 bare
2 water
0 ice
Slovakia
44 crop
49 tree
4 grass
0 flooded
2 urban
0 bare
1 water
Ukraine
79 crop
13 tree
2 grass
0 flooded
2 urban
0 bare
3 water
0 ice
Hungary
72 crop
17 tree
7 grass
1 flooded
2 urban
0 bare
1 water
Austria
23 crop
57 tree
15 grass
0 flooded
1 urban
2 bare
1 water
France
51 crop
28 tree
16 grass
0 flooded
2 urban
0 bare
4 water
0 ice
Switzerland
16 crop
41 tree
29 grass
0 flooded
2 urban
6 bare
3 water
2 ice
Croatia
42 crop
47 tree
5 grass
1 flooded
1 urban
0 bare
5 water
Romania
58 crop
32 tree
7 grass
1 flooded
2 urban
0 bare
1 water
Moldova
88 crop
7 tree
0 grass
0 flooded
3 urban
0 bare
1 water
0 ice
Slovenia
25 crop
70 tree
3 grass
0 flooded
1 urban
Italy
51 crop
31 tree
6 grass
0 flooded
2 urban
3 bare
7 water
0 ice
Bosnia
27 crop
62 tree
10 grass
Kosovo
49 crop
44 tree
6 grass
0 flooded
1 urban
Serbia
62 crop
33 tree
4 grass
0 flooded
1 urban
0 bare
1 water
Bulgaria
59 crop
36 tree
2 grass
0 flooded
1 urban
1 bare
1 water
Georgia
37 crop
51 tree
8 grass
0 flooded
1 urban
1 bare
2 water
0 ice
Spain
53 crop
35 tree
8 grass
0 flooded
1 urban
0 bare
3 water
0 ice
Montenegro
21 crop
66 tree
11 grass
1 flooded
1 urban
0 bare
0 water
0 ice
Albania
52 crop
32 tree
7 grass
0 flooded
1 urban
2 bare
6 water
0 ice
Greece
40 crop
38 tree
4 grass
0 flooded
1 urban
1 bare
15 water
0 ice
Cyprus
37 crop
32 tree
10 grass
0 flooded
1 urban
8 bare
12 water
0 ice
Turkey
55 crop
24 tree
14 grass
0 flooded
0 urban
2 bare
5 water
0 ice
Armenia
54 crop
26 tree
12 grass
0 flooded
1 urban
2 bare
4 water
0 ice
Azerbaijan
58 crop
18 tree
14 grass
0 flooded
0 urban
5 bare
4 water
0 ice
Macedonia
44 crop
42 tree
12 grass
0 flooded
1 urban
0 bare
2 water
0 ice
Lebanon
58 crop
22 tree
10 grass
0 flooded
1 urban
5 bare
4 water
0 ice
Syria
29 crop
1 tree
5 grass
0 flooded
0 urban
64 bare
1 water
0 ice
Iraq
22 crop
0 tree
3 grass
0 flooded
0 urban
73 bare
2 water
0 ice
Iran
18 crop
5 tree
13 grass
0 flooded
0 urban
64 bare
1 water
0 ice
Morocco
15 crop
6 tree
15 grass
0 flooded
0 urban
61 bare
2 water
0 ice
Algeria
4 crop
1 tree
4 grass
0 flooded
0 urban
90 bare
0 water
0 ice
Tunisia
28 crop
3 tree
12 grass
0 flooded
0 urban
54 bare
3 water
0 ice
Palestine
49 crop
7 tree
20 grass
0 flooded
2 urban
21 bare
2 water
0 ice
Israel
34 crop
3 tree
5 grass
0 flooded
2 urban
54 bare
3 water
0 ice
Jordan
6 crop
0 tree
2 grass
0 flooded
0 urban
91 bare
1 water
0 ice
Kuwait
2 crop
0 tree
2 grass
0 flooded
1 urban
89 bare
6 water
0 ice
Qatar
2 crop
1 tree
1 grass
0 flooded
3 urban
85 bare
Libya
1 crop
0 tree
2 grass
0 flooded
Egypt
5 crop
0 tree
1 grass
v3.indd 51
20/01/2024 14:44




Western Sahara
0 crop
0 tree
0 grass
0 flooded
0 urban
100 bare
0 water
0 ice
Mauritania
2 crop
1 tree
15 grass
0 flooded
0 urban
81 bare
1 water
0 ice
Mali
20 crop
10 tree
14 grass
0 flooded
0 urban
56 bare
0 water
0 ice
Niger
9 crop
1 tree
23 grass
1 flooded
0 urban
67 bare
0 water
0 ice
Senegal
30 crop
56 tree
7 grass
3 flooded
0 urban
0 bare
4 water
0 ice
Gambia
60 crop
20 tree
0 grass
10 flooded
0 urban
0 bare
9 water
0 ice
Guinea
7 crop
83 tree
1 grass
0 flooded
1 urban
0 bare
0 water
10 ice
Burkina Faso
68 crop
14 tree
17 grass
0 flooded
0 urban
0 bare
0 water
0 ice
Guinea-Bissau
20 crop
53 tree
0 grass
11 flooded
0 urban
0 bare
15 water
0 ice
Côte d'Ivoire
49 crop
49 tree
0 grass
0 flooded
0 urban
0 bare
2 water
0 ice
Ghana
45 crop
50 tree
0 grass
0 flooded
0 urban
0 bare
4 water
0 ice
Nigeria
60 crop
34 tree
4 grass
1 flooded
0 urban
0 bare
2 water
0 ice
Cameroon
14 crop
83 tree
0 grass
1 flooded
0 urban
0 bare
2 water
0 ice
Sierra Leone
63 crop
28 tree
0 grass
3 flooded
0 urban
0 bare
5 water
0 ice
Liberia
43 crop
50 tree
0 grass
0 flooded
0 urban
0 bare
7 water
0 ice
Togo
49 crop
48 tree
0 grass
0 flooded
0 urban
0 bare
3 water
0 ice
Benin
36 crop
62 tree
0 grass
0 flooded
0 urban
0 bare
2 water
0 ice
Equatorial Guinea
7 crop
83 tree
0 grass
1 flooded
0 urban
0 bare
10 water
0 ice
Gabon
4 crop
89 tree
1 grass
1 flooded
0 urban
0 bare
4 water
0 ice

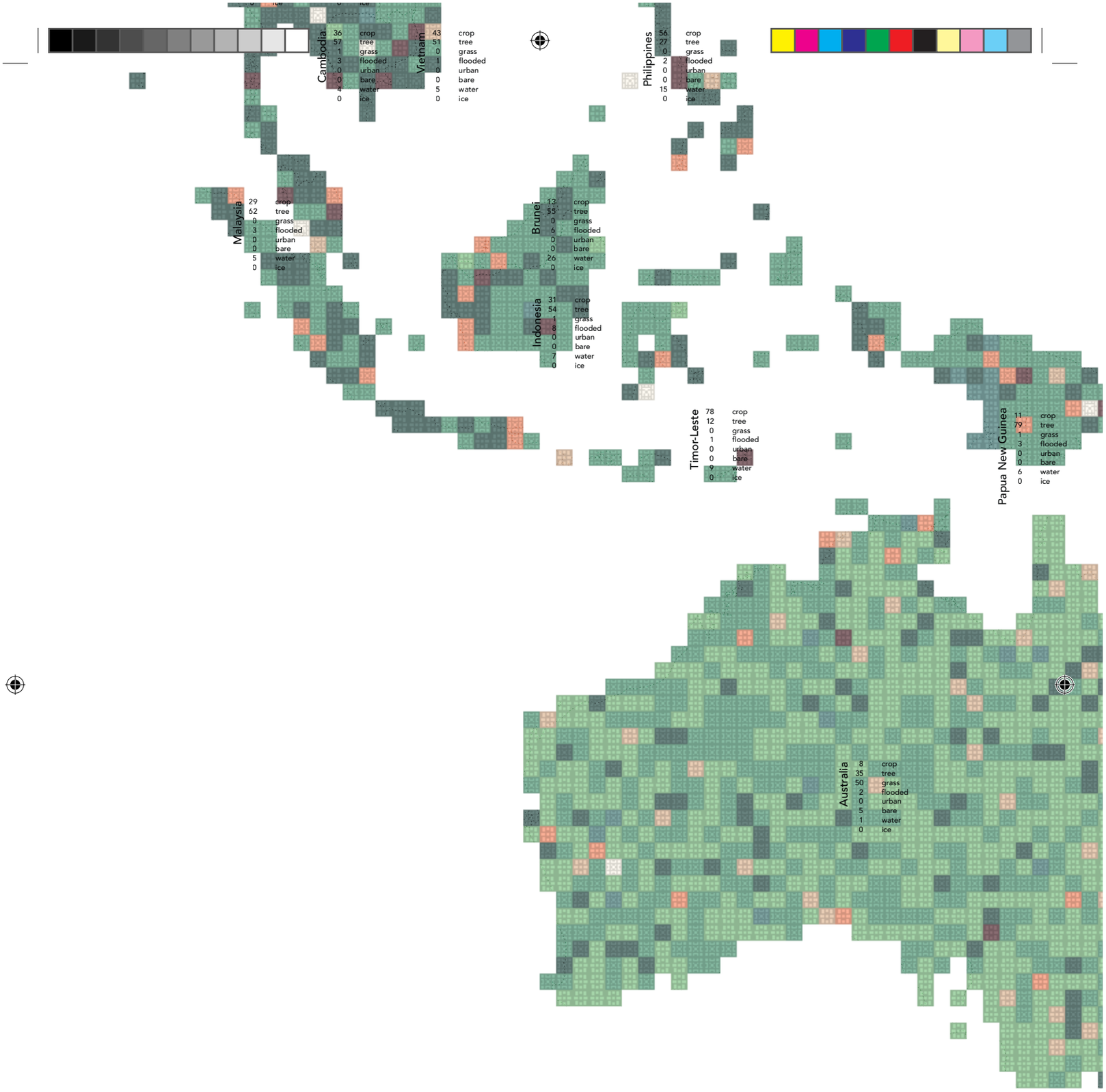

Libya
1 crop
0 tree
2 grass
0 flooded
0 urban
96 bare
0 water
0 ice
Egypt
5 crop
0 tree
1 grass
1 flooded
0 urban
91 bare
2 water
0 ice
Kuwait
2 crop
0 tree
2 grass
0 flooded
1 urban
89 bare
6 water
0 ice
Qatar
2 crop
1 tree
1 grass
0 flooded
3 urban
85 bare
8 water
0 ice
Saudi Arabia
1 crop
0 tree
4 grass
0 flooded
0 urban
94 bare
0 water
0 ice
Oman
1 crop
0 tree
6 grass
0 flooded
0 urban
92 bare
1 water
0 ice
United Arab Emirates
3 crop
0 tree
4 grass
0 flooded
2 urban
87 bare
4 water
0 ice
Yemen
4 crop
4 tree
11 grass
0 flooded
0 urban
78 bare
2 water
0 ice
Djibouti
0 crop
1 tree
4 grass
4 flooded
0 urban
87 bare
3 water
0 ice
Niger
9 crop
1 tree
23 grass
1 flooded
0 urban
67 bare
0 water
0 ice
Chad
18 crop
15 tree
16 grass
0 flooded
0 urban
51 bare
0 water
0 ice
Sudan
5 crop
86 tree
3 grass
6 flooded
0 urban
0 bare
0 water
0 ice
Eritrea
20 crop
18 tree
20 grass
0 flooded
0 urban
40 bare
2 water
0 ice
Ethiopia
25 crop
52 tree
6 grass
0 flooded
0 urban
7 bare
1 water
0 ice
Cameroon
14 crop
83 tree
0 grass
1 flooded
0 urban
0 bare
2 water
0 ice
Central African Republic
2 crop
98 tree
0 grass
0 flooded
0 urban
0 bare
0 water
0 ice
Equatorial Guinea
7 crop
83 tree
0 grass
1 flooded
0 urban
0 bare
10 water
0 ice
Uganda
49 crop
30 tree
3 grass
2 flooded
0 urban
0 bare
16 water
0 ice
Kenya
24 crop
53 tree
14 grass
2 flooded
0 urban
5 bare
2 water
0 ice
Somalia
12 crop
68 tree
12 grass
0 flooded
0 urban
4 bare
3 water
0 ice
Gabon
4 crop
89 tree
1 grass
1 flooded
0 urban
0 bare
4 water
0 ice
Congo [Republic]
13 crop
72 tree
3 grass
12 flooded
0 urban
0 bare
1 water
0 ice
Congo [DRC]
11 crop
77 tree
1 grass
9 flooded
0 urban
0 bare
2 water
0 ice
Rwanda
71 crop
24 tree
0 grass
2 flooded
0 urban
0 bare
3 water
0 ice
Burundi
57 crop
23 tree
12 grass
1 flooded
0 urban
0 bare
8 water
0 ice
Tanzania
28 crop
56 tree
3 grass
5 flooded
0 urban
0 bare
8 water
0 ice
Angola
6 crop
77 tree
12 grass
1 flooded
0 urban
2 bare
1 water
0 ice
Zambia
15 crop
78 tree
1 grass
4 flooded
0 urban
0 bare
2 water
0 ice
Malawi
44 crop
39 tree
2 grass
1 flooded
0 urban
0 bare
13 water
0 ice
Mozambique
13 crop
79 tree
3 grass
2 flooded
0 urban
1 bare
3 water
0 ice
Namibia
3 crop
29 tree
48 grass
0 flooded
0 urban
19 bare
1 water
0 ice
Botswana
5 crop
85 tree
6 grass
1 flooded
0 urban
2 bare
0 water
0 ice
Zimbabwe
36 crop
58 tree
4 grass
0 flooded
0 urban
0 bare
1 water
0 ice
Madagascar
12 crop
83 tree
1 grass
1 flooded
0 urban
0 bare
2 water
0 ice
Swaziland
27 crop
56 tree
15 grass
1 flooded
0 urban
1 bare
0 water
0 ice
South Africa
13 crop
60 tree
25 grass
0 flooded
0 urban
0 bare
1 water
0 ice
Lesotho
24 crop
41 tree
35 grass
0 flooded
0 urban
0 bare
0 water
0 ice

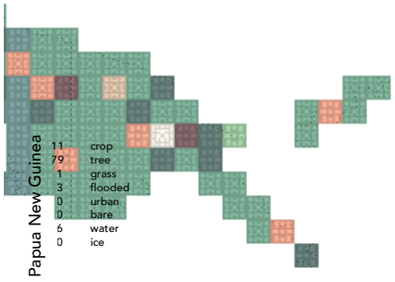


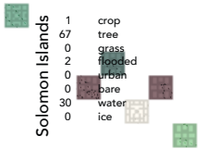

Russia
11 crop
63 tree
14 grass
5 flooded
0 urban
2 bare
4 water
0 ice
Mongolia
6 crop
7 tree
42 grass
0 flooded
0 urban
45 bare
1 water
0 ice
China
29 crop
20 tree
30 grass
0 flooded
0 urban
18 bare
2 water
1 ice
North Korea
31 crop
65 tree
1 grass
0 flooded
0 urban
0 bare
3 water
0 ice
South Korea
35 crop
54 tree
1 grass
0 flooded
1 urban
1 bare
8 water
0 ice
Japan
22 crop
60 tree
0 grass
0 flooded
3 urban
0 bare
15 water
0 ice
31 crop
62 tree

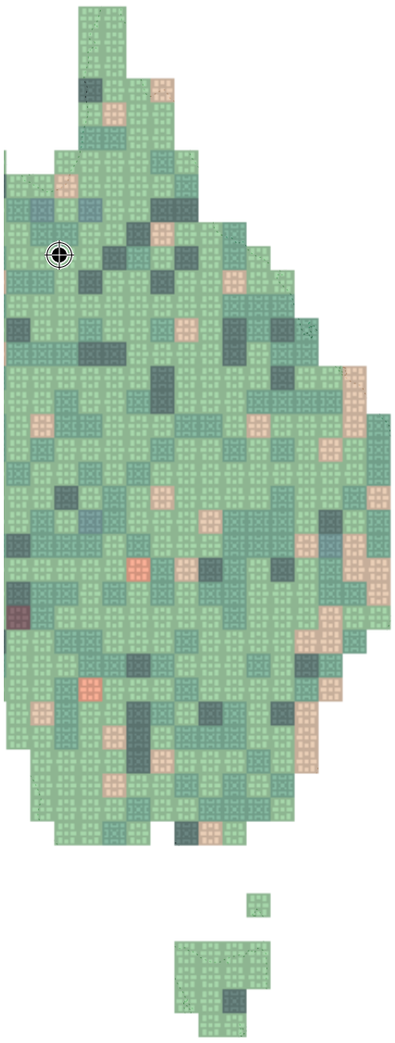

Israel
Jordan
Iraq
Kuwait
Saudi Arabia
Yemen
Djibouti
Eritrea
Ethiopia
Uganda
Kenya
Somalia
Burundi
Tanzania
Malawi
Mozambique
Madagascar
Iran
Qatar
United Arab Emirates
Oman
Afghanistan
Pakistan
Nepal
Bhutan
India
Bangladesh
Sri Lanka
crop
tree
grass
flooded
urban
bare
water
ice

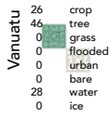

Bhutan
3 crop
81 tree
11 grass
0 flooded
0 urban
0 bare
0 water
3 ice
Bangladesh
74 crop
13 tree
1 grass
3 flooded
0 urban
0 bare
9 water
0 ice
Myanmar [Burma]
32 crop
63 tree
1 grass
1 flooded
0 urban
0 bare
3 water
0 ice
Thailand
67 crop
29 tree
0 grass
1 flooded
0 urban
0 bare
3 water
0 ice
Laos
17 crop
82 tree
0 grass
0 flooded
0 urban
0 bare
1 water
0 ice
Cambodia
36 crop
57 tree
1 grass
3 flooded
0 urban
0 bare
4 water
0 ice
Vietnam
43 crop
51 tree
0 grass
1 flooded
0 urban
0 bare
5 water
0 ice
Taiwan
31 crop
62 tree
0 grass
2 flooded
2 urban
0 bare
4 water
0 ice
Philippines
56 crop
27 tree
0 grass
2 flooded
0 urban
0 bare
15 water
0 ice
Malaysia
29 crop
62 tree
0 grass
3 flooded
0 urban
0 bare
5 water
0 ice
Brunei
13 crop
55 tree
0 grass
6 flooded
0 urban
0 bare
26 water
0 ice
Indonesia
31 crop
54 tree
1 grass
8 flooded
0 urban
0 bare
7 water
0 ice
Timor-Leste
78 crop
12 tree
0 grass
1 flooded
0 urban
0 bare
9 water
0 ice
Papua New Guinea
11 crop
79 tree
1 grass
3 flooded
0 urban
0 bare
6 water
0 ice
Australia
8 crop
35 tree
50 grass
2 flooded
0 urban
5 bare
1 water
0 ice

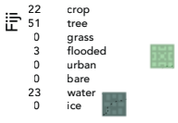

Cambodia
36 crop
57 tree
1 grass
3 flooded
0 urban
0 bare
4 water
0 ice
Vietnam
43 crop
51 tree
0 grass
1 flooded
0 urban
0 bare
5 water
0 ice
Philippines
56 crop
27 tree
0 grass
2 flooded
0 urban
0 bare
15 water
0 ice
Malaysia
29 crop
62 tree
0 grass
3 flooded
0 urban
0 bare
5 water
0 ice
Brunei
13 crop
55 tree
0 grass
6 flooded
0 urban
0 bare
26 water
0 ice
Indonesia
31 crop
54 tree
1 grass
8 flooded
0 urban
0 bare
7 water
0 ice
Timor-Leste
78 crop
12 tree
0 grass
1 flooded
0 urban
0 bare
9 water
0 ice
Papua New Guinea
11 crop
79 tree
1 grass
3 flooded
0 urban
0 bare
6 water
0 ice
Australia
8 crop
35 tree
50 grass
2 flooded
0 urban
5 bare
1 water
0 ice

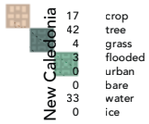


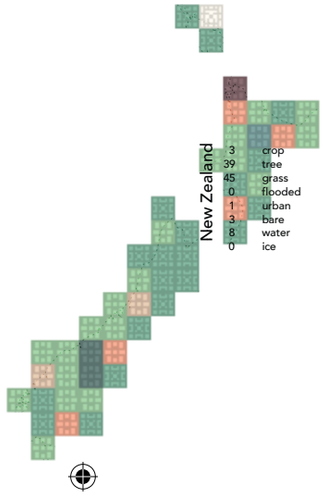












Screen printing, also known as silk screening, is a versatile and popular printing technique used to transfer images onto various surfaces such as fabric, paper, glass, metal, and more. It's widely employed in the creation of T-shirts, posters, signage, and other promotional items.

Design Preparation:

Begin with a design or image that you want to print.

Convert the design into a stencil or screen, usually using a light-sensitive emulsion coated onto a mesh screen.

Screen Preparation:

The mesh screen is typically made of polyester or other materials.

The screen is coated with a light-sensitive emulsion that hardens when exposed to light.

Image Exposure:

Place the stencil or design on the coated screen.

Expose the screen to light, causing the emulsion to harden in the areas not covered by the design.

Rinsing:

Rinse the screen to remove the unexposed emulsion, leaving behind the open areas corresponding to the design.

Printing Setup:

Position the screen on the printing surface (e.g., fabric, paper).

Use a squeegee to push ink through the open areas of the screen onto the substrate, transferring the design.

Printing Process:

Place the substrate beneath the screen.

Apply the ink to the screen's top edge and use the squeegee to evenly spread it over the design area.

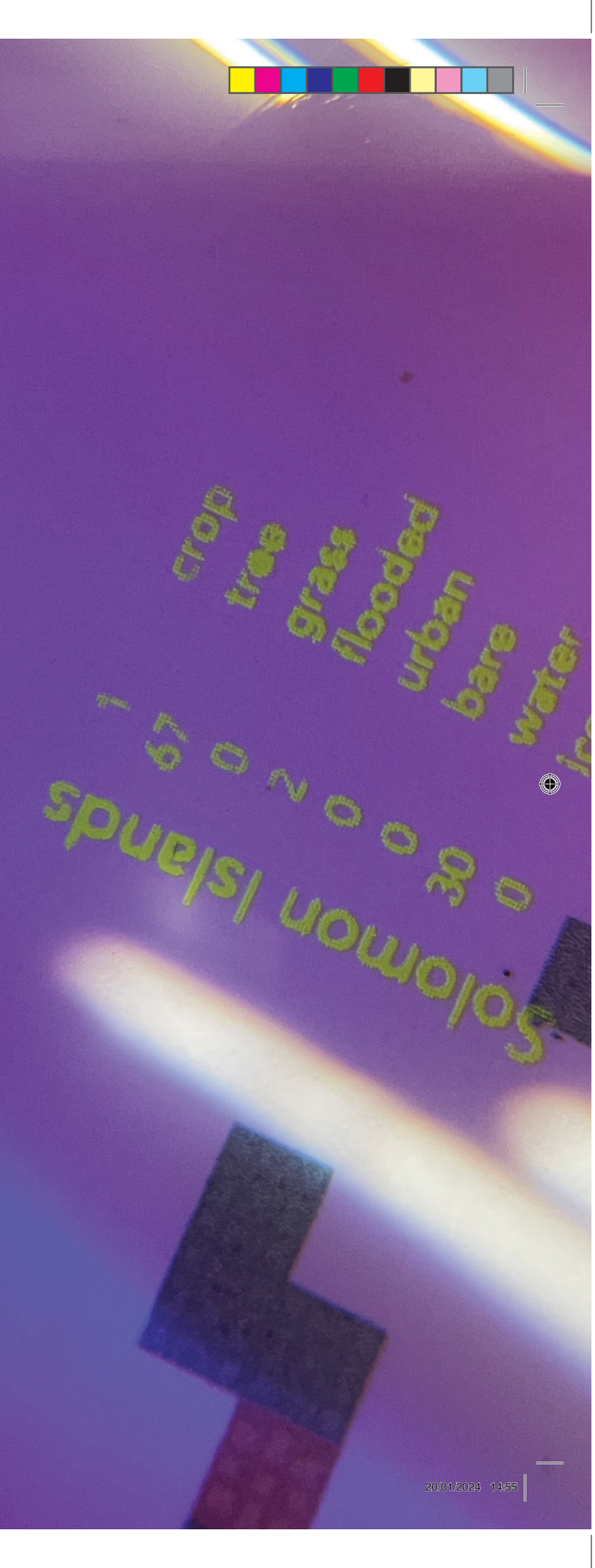
Curing:

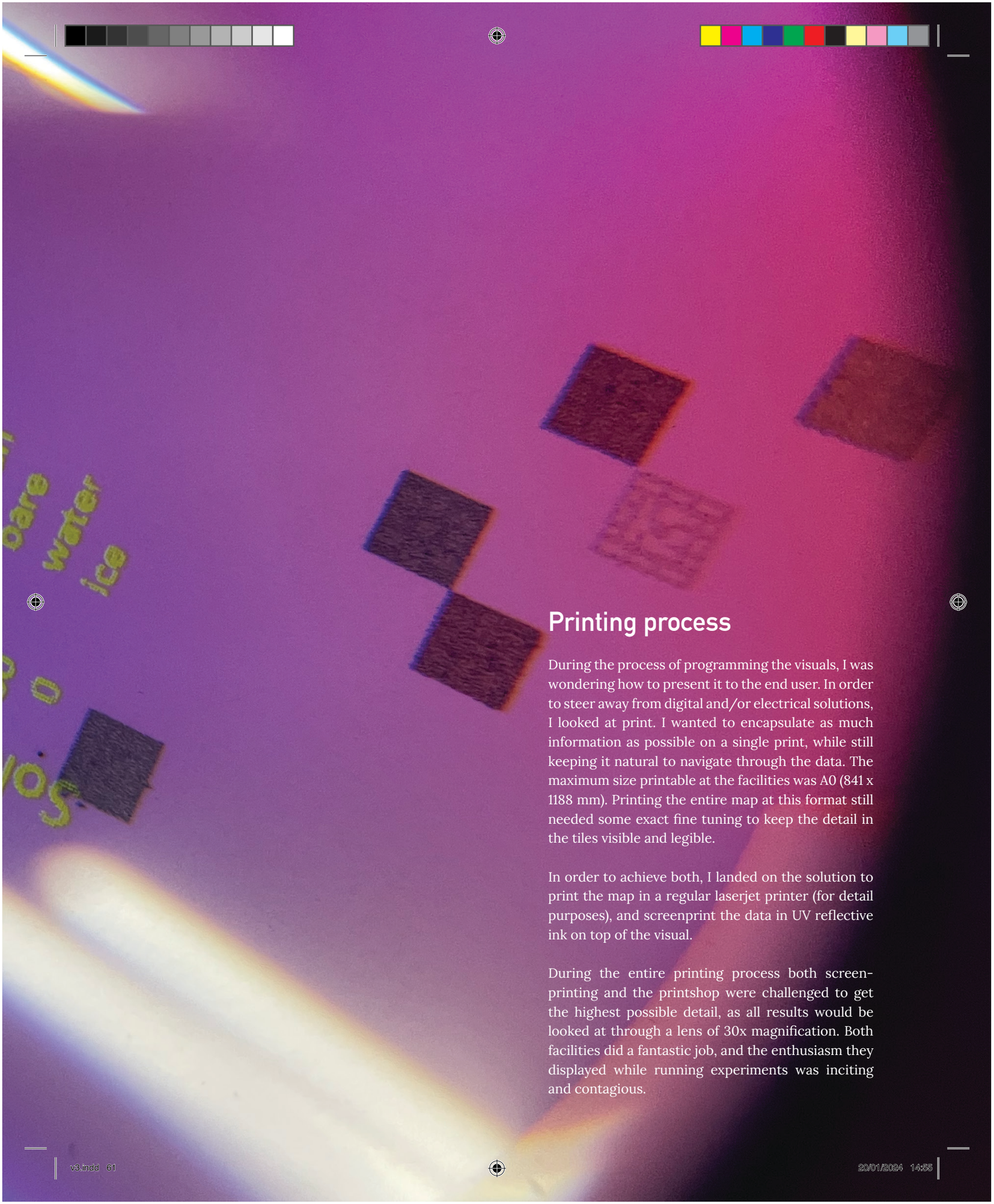
Cure or dry the printed material, usually through heat or air drying, to ensure the ink adheres properly.

Repeat:

If multiple colors are involved, the process is repeated for each color layer, allowing for intricate and multi-colored designs.

Screen printing offers several advantages, including high-quality and durable prints, the ability to print on various materials, and cost-effectiveness for large production runs. It's a popular choice for both small-scale artists and large-scale commercial printing operations.





Printing process

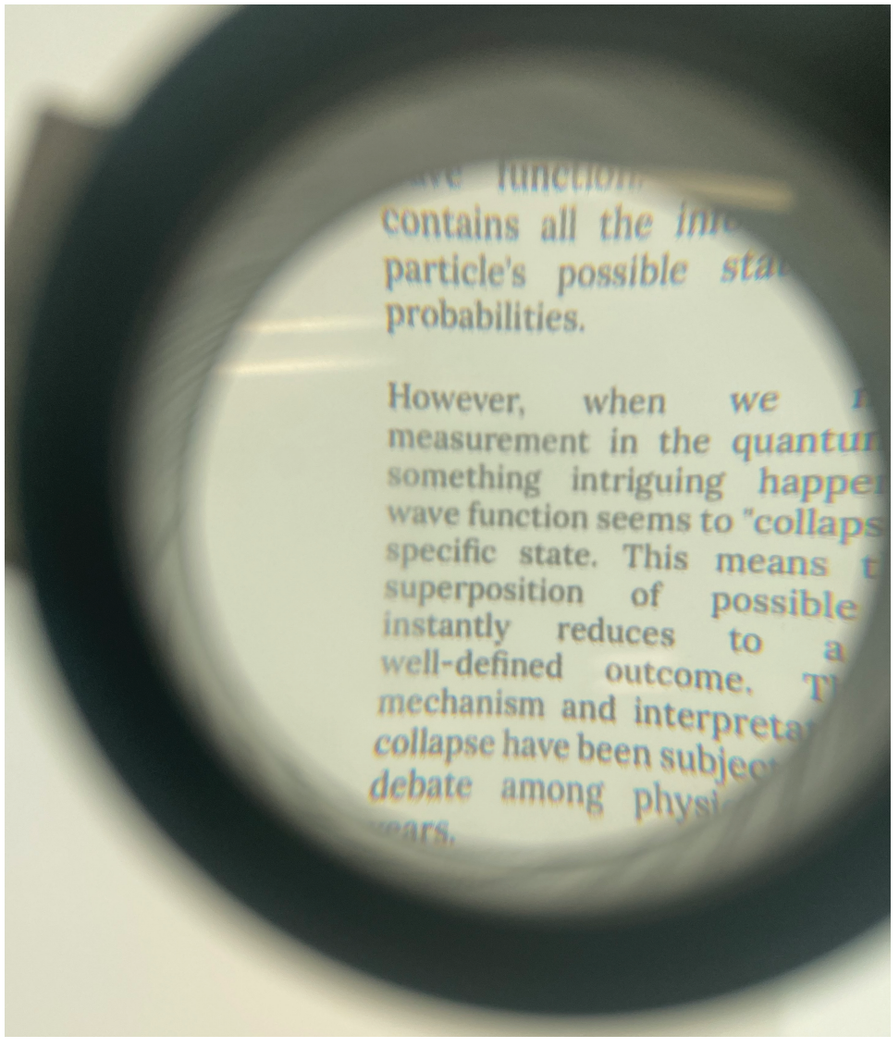
During the process of programming the visuals, I was wondering how to present it to the end user. In order to steer away from digital and/or electrical solutions, I looked at print. I wanted to encapsulate as much information as possible on a single print, while still keeping it natural to navigate through the data. The maximum size printable at the facilities was A0 (841 x 1188 mm). Printing the entire map at this format still needed some exact fine tuning to keep the detail in the tiles visible and legible.

In order to achieve both, I landed on the solution to print the map in a regular laserjet printer (for detail purposes), and screenprint the data in UV reflective ink on top of the visual.

During the entire printing process both screen-printing and the printshop were challenged to get the highest possible detail, as all results would be looked at through a lens of 30x magnification. Both facilities did a fantastic job, and the enthusiasm they displayed while running experiments was inciting and contagious.

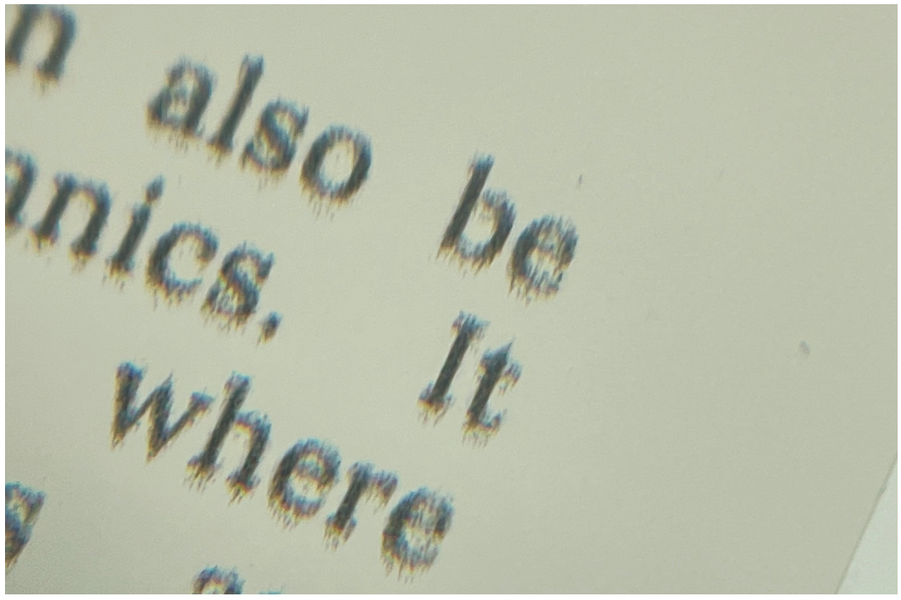
Printing process

During the concept phase, I was wondering how I could add the data to the visual map without influencing the print underneath, and add a little extra to the project. I had a few options; either I could print very big, and make the text legible overall. This way, I could print the project on A0, on the standard available printers in the LUCA print lab. However, the possibility was suggested to me to print this in UV reflective ink, being invisible to the naked eye, but can emerge as a puzzle piece when lit with black light.



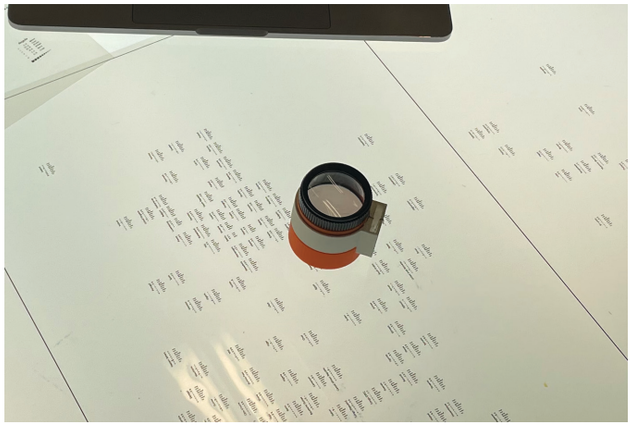
I wandered into the print lab in LUCA and asked the experienced coordinator if this was a possibility for the machinery available. Sadly, it wasn't, but Maarten was kind enough to point me to the printmaking (screen printing) workshop one flight above the print shop. I had no idea it was an option to walk into the shop and started printing, so I was pleasantly surprised to find this openness.

My first question was (regarding my digital background) about the possible



resolution of the screen-printing process. The finest sieve available counts 120 wires per centimeter, which is not really the final resolution (in contrast to the logic). The detail with which one can print is influenced by the blocking emulsion, the ink viscosity and other factors. Because of these reasons, the answer of the expert present in the lab was "I don't know, let's find out", which gave an impression of the passion displayed in these workshop.

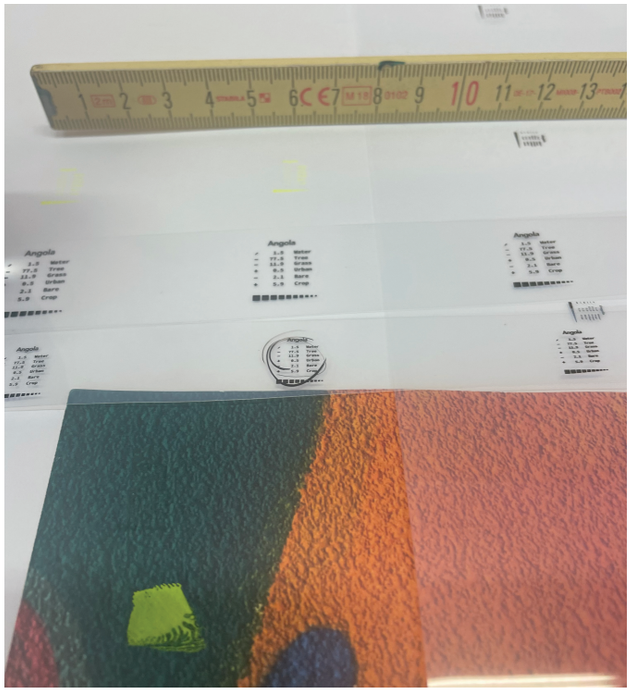
The next hours were spent running



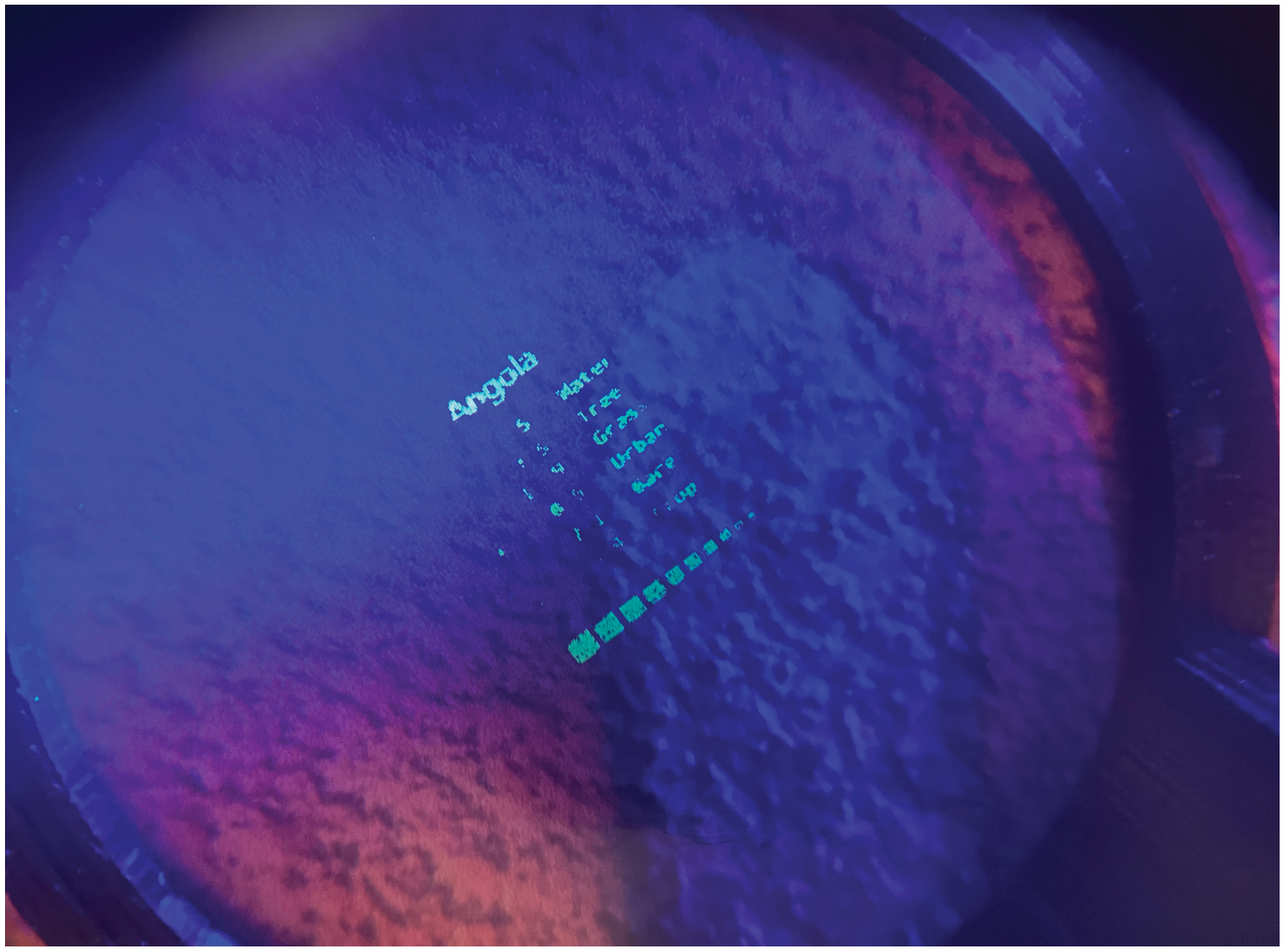
down to the print shop, designing a quick iteration sheet for the detail capturing (an A4 with incrementally shrinking design elements) visible on the right here, and printing these on a transparent sheet to prep it for transfer to a sheet. Mind you that the quality of the laser print on the sheet is of vital importance as well. Any deviation on the transfer sheet will show up on the screen, and thus in the print. Eventually, the absolute expertise of Maarten (print shop) made the laser-printer spit out details I previously deemed impossible. I was looking at the prints through a 30x magnification lens, here above. On the final prints we used for the screen printing, the data was printed on about a centimeter wide, and was clearly legible through this lens, without any artifacts visible of the printing process (right).

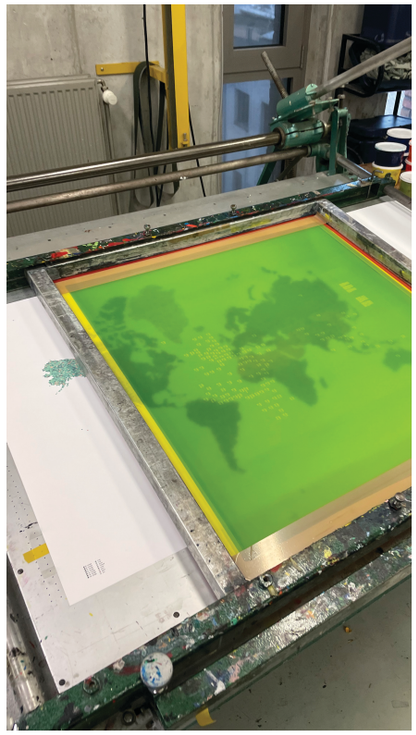
We settled on a size that was still legible when printed on a (scrap) colored surface, but only just (and only readable without the lens when really getting close). This was the baseline for the entire rest of the process.

Elke Van Kerckvoorde guided me through this process with the necessary enthusiasm and passion in the printmaking process, pointing out opportunities and pitfalls, thinking with me about the process and the possible configurations of the screens. In terms of paper, I was suggested a format of 72x102 cm, arena smooth 170gr.



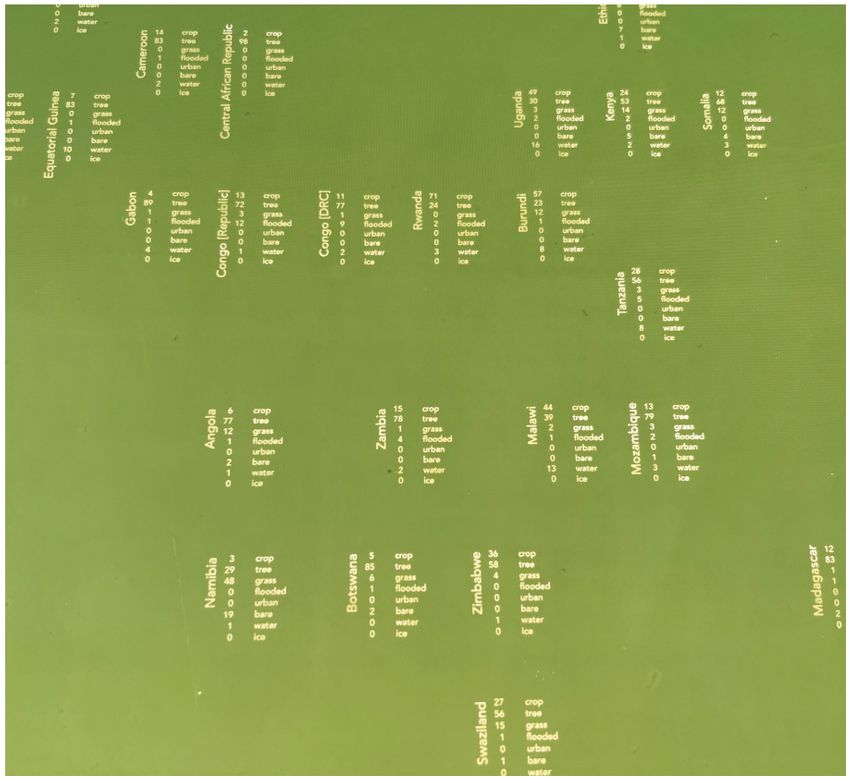
The overlaid data sheets were split into 3 different sheets of size A3, arranged on two screens. On the day of printing, I went to the process of printing a few of the base prints (3 in total), and the transfer sheets. I washed the screens, prepared them (for the first time in my life), and blasted the emulsion with UV to cure it.





Printing this small requires expertise I frankly don;t have (yet), so my small errors were visible in the print, and I managed to clog up an entire screen at some point. But in the end, I had 3 printed sheets, and a fourth print on an empty sheet for reference.

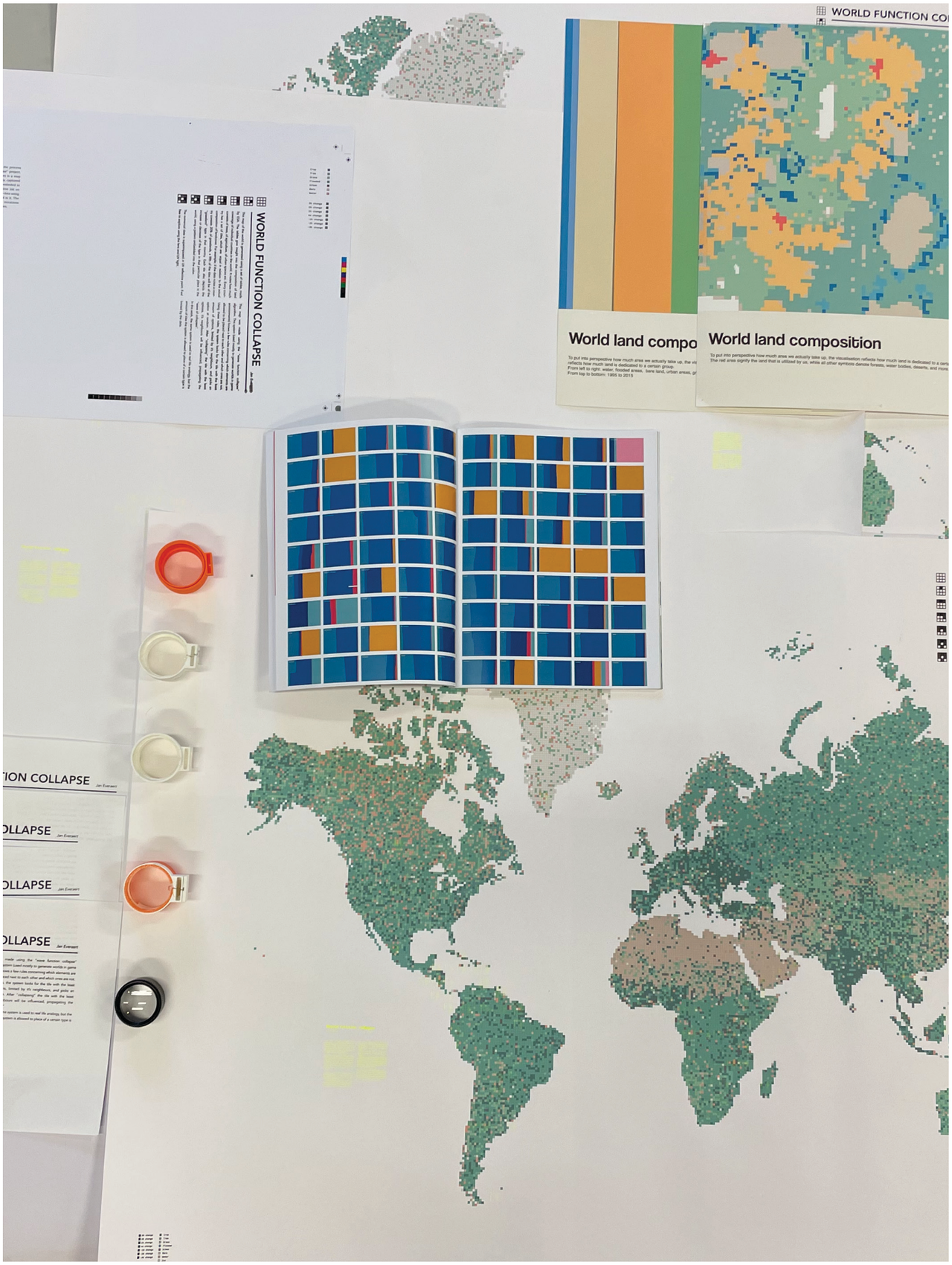
I've reused the film since for a first iteration of the cover of this print work, and added a header to the top part of the map in order to better explain to a non-informed visitor what's going on.



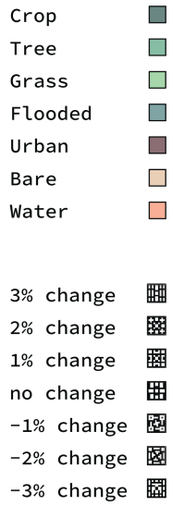
The 3D printed holder for the lens (on the right in the picture above) was constructed to hold the lens at the optimal height, to ensure the biggest enhancement The bottom of the lens holder had to be 21mm up from the page.

Besides the offset, the holder also servers to hold the UV light, and coin battery to illuminate the UV print (hence the purple light when pictures are taken through the lens).

In the early stages of the concept, I was considering showing the data with a positioning system, where depending on the location of the lens on the print, different data would be shown on a display. I was motivated by my lecturers to steer away from overly technical solutions, and fixate on the design aspects.



I would like to extend my greatest thanks to all staff and involved students in this work. I have greatly appreciated and enjoyed the expertise and enthusiasm.



Book Cover by Jan Everaert

Illustrations and data visualizations by Jan Everaert

First edition 2024

An extensive look into the creation of a world map, based on ESA data. Three months of research and experimentation with print, processing and land composition data.