QGIS UC 2023 workshop

“Create your own promotion film in QGIS”

Trainers: Quinten Dengerink & Derek van Bochove (Geon)

18-4-2023

In this workshop, you will create a promotion film, using QGIS. This film (animated map) will show every user conference until now. We will do the necessary work in QGIS step by step.

Contents

1. Load data 2

2. Generate line with the geometry generator 3

2a. Generate a line from each location to our current location 3

2b. Generate a curvy line 4

3. Visualisation line 7

4. Visualisation basemap 9

5. Temporal controller 10

5.1 Turn on Temporal Settings of layer 10

5.2 Temporal Controller Panel 10

5.3 Add timed labels 11

6. Map decoration 12

7. Export 14

8. Make a movie out of the data 15

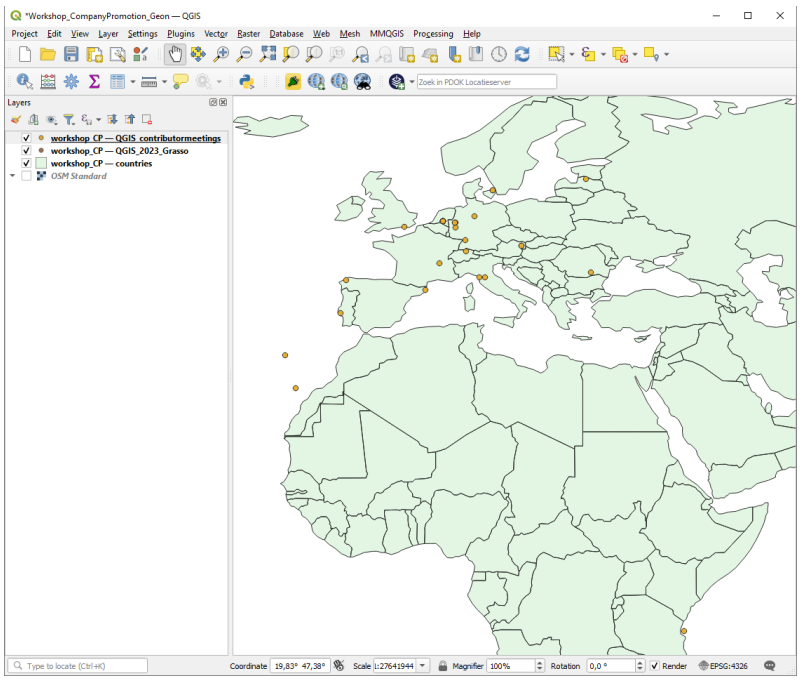
1. Load data

We're going to visualise the lines from previous QGIS contributor meetings and hackfests going towards today's venue, Grasso in Den Bosch.

- Start an empty QGIS project
- Load the 3 layers QGIS_contributormeetings, QGIS_2023_Grasso and countries from the geopackage Workshop_CP

This can also be done for a map of your clients from your company or organisation. In that case you need one file with the locations (points) of your clients, including a date, and one file with the location (point) of your company or organisation.

- Add for a reference a base map to your liking, e.g. by using the QuickMapService plugin.



2. Generate line with the geometry generator

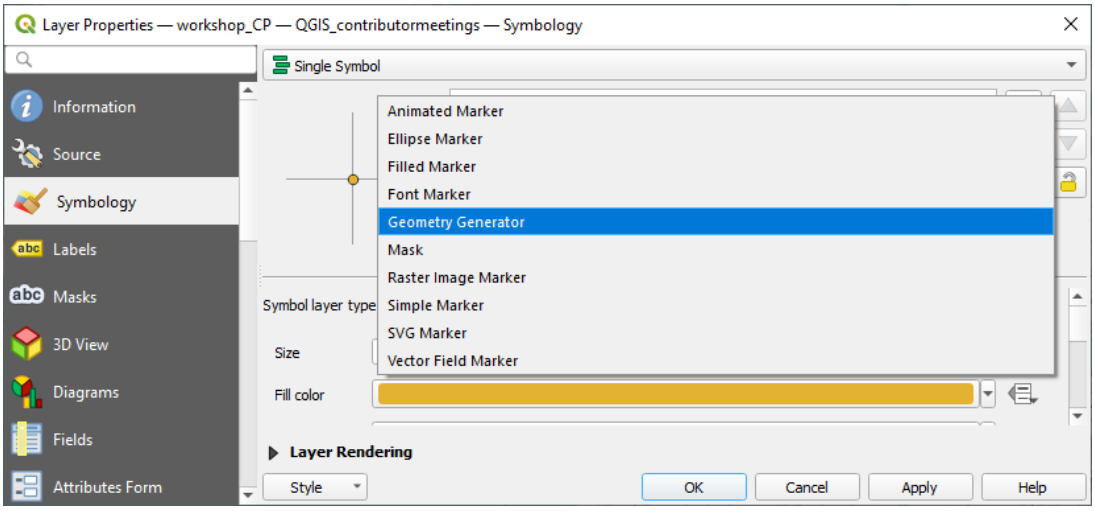
In this chapter, we're going to generate lines from each point to our location in Den Bosch.

Don't be afraid of a little bit of QGIS expressions! Let's do it step by step.

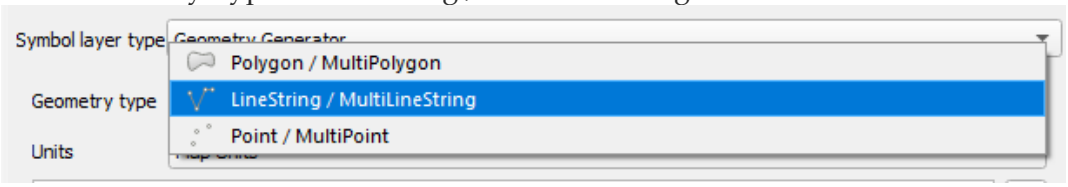
2a. Generate a line from each location to our current location

First, let's generate a line from each point to our location in Den Bosch.

- Open the Symbology window of the QGIS_contributormeetings layer.
- Activate the geometry generator (change Simple Marker to Geometry Generator):



- Set Geometry Type on Linestring / MultiLineString



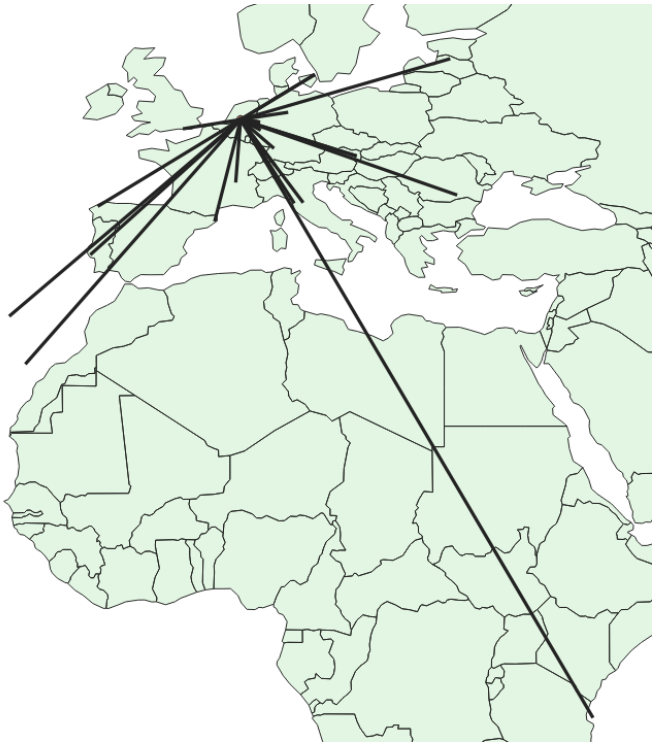
Now it is time for QGIS expressions!

- Open the *Expression Dialog*
- Use **`make_line()`** to generate a line from the layer's own geometry to the geometry of our location in Den Bosch.

> We can use **`$geometry`** to get the geometry of each event and use **`get_feature_by_id()`** with the layer id of the QGIS_2023_Grasso layer and the fid as arguments. This results in the following expression:

```
make_line( @geometry,
geometry(                                get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)))
```

Note: Use *Map Layers* in the *Expression Dialog* for the right layer id in this expression.



2b. Generate a curvy line

Let's turn the line that we have into a curvy line. That will look a bit more exciting. For that, we will need the symbol layer type 'Arrow'. But, at the moment we have completely straight lines between two points. For QGIS, it's impossible to show a curve between two points. For that, we will need a centroid that is offset from the straight line.

Let's first generate the centroid of the straight line.

- Use **`centroid()`** on the existing expression to create the centroid.

```
centroid(    make_line(    @geometry,    geometry(    get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)
)))
```

Now that we have found the centroid of this line, we can make a new line between the event location, the centroid and our location. We will need to combine the expressions that we already have.

- Add the **centroid()** expression as second geometry in the **make_line()** geometry. This will result in the following expression:

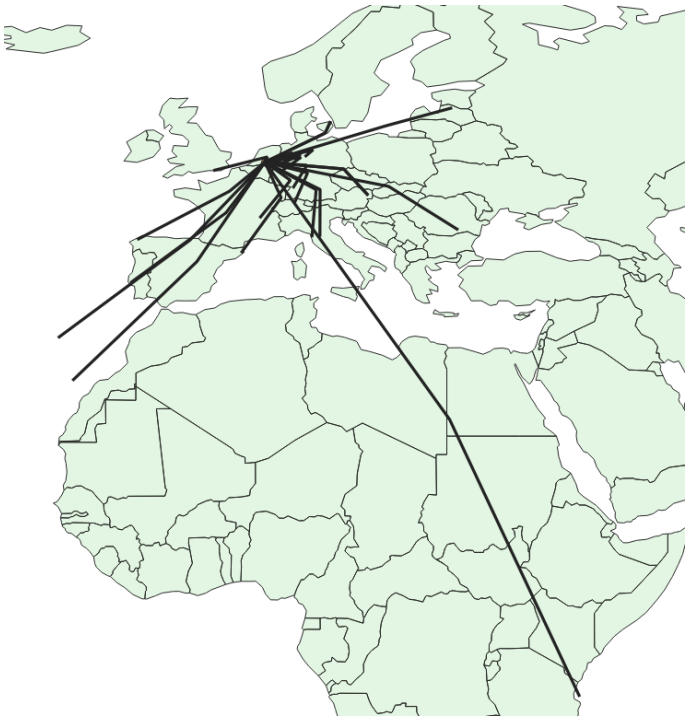
```
make_line(@geometry,
centroid(        make_line(@geometry,        geometry(get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)
))),
geometry(get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)
))
```

This will result in the same straight line. It does not look different from the straight line that we already had. We are going to offset the centroid a little bit, by using the **translate()** expression.

- Wrap the `translate()` expression around the centroid, and give it the arguments 1 and 1. This will translate (move) the centroid by 1 degree on the X-axis and 1 degree on the Y-axis. In this case degrees because we are using WGS84 as coordinate system. This will result in the following expression:

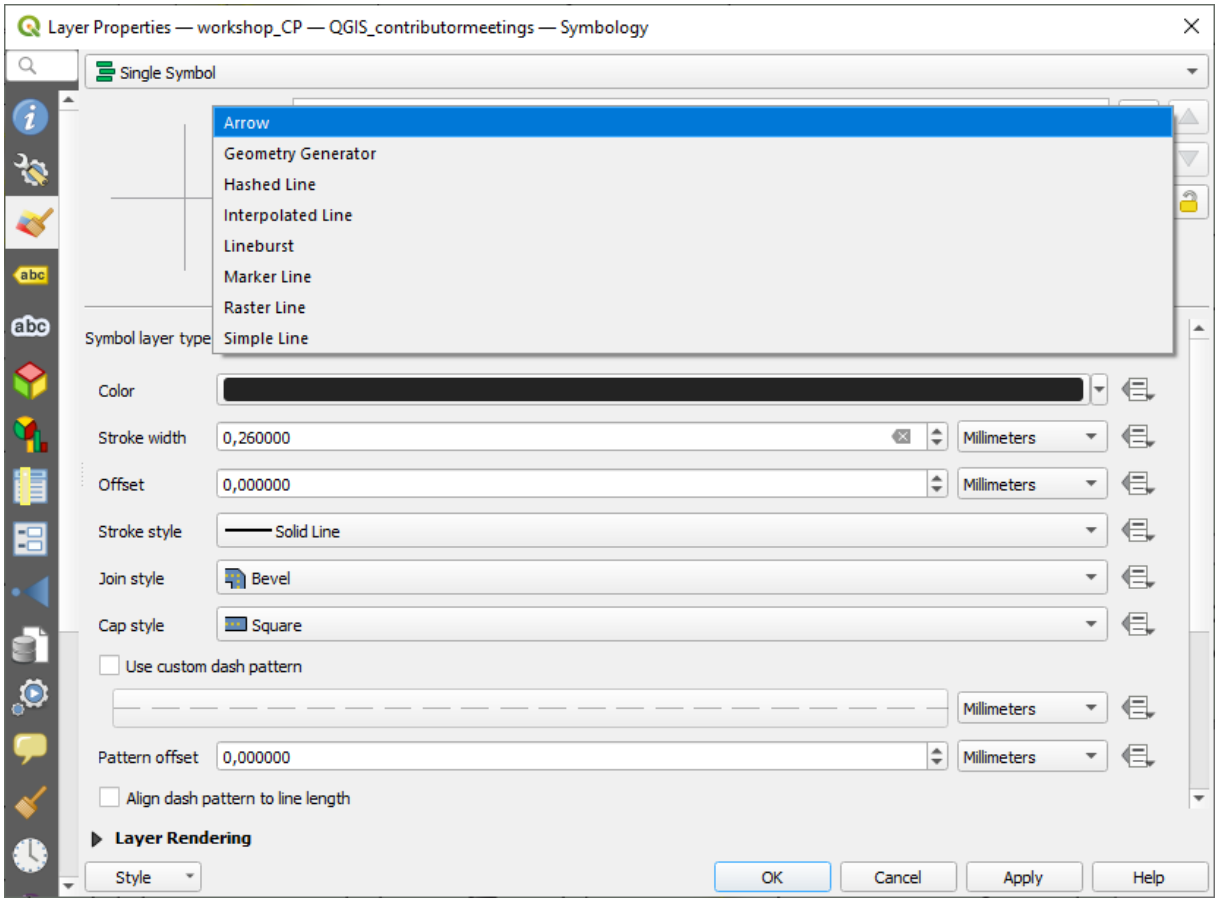
```
make_line(@geometry,
translate( centroid( make_line(  @geometry, geometry( get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)))
), 2,2),
 geometry(                                              get_feature_by_id(
'workshop_CP___QGIS_2023_Grasso_4cc6f2e7_5ab7_4ea4_827b_ab9ad4fab2a9',1)))
```

And it will look like this:



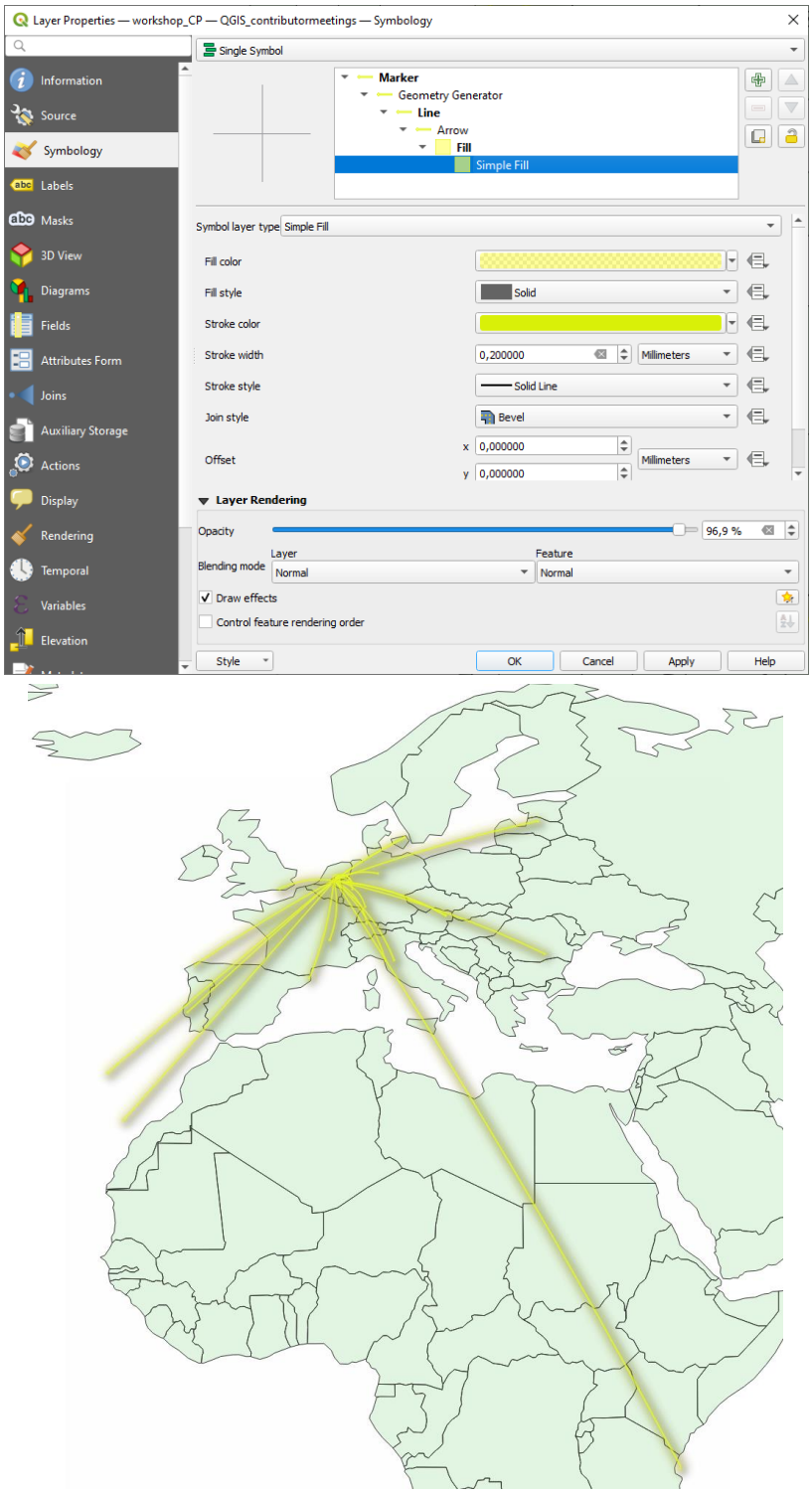
3. Visualisation line

The line does not look curvy yet! We're going to use the Arrow symbol layer type instead of the Simple Line, to make it curvy.



- Select Arrow

Change the settings according to your own taste! But make sure the box next to "Curved Arrows" is ticked. Light, bright colors are recommended.


Layer Properties — workshop_CP — QGIS_contributormeetings — Symbology
Information
Source
Symbology
Labels
Masks
3D View
Diagrams
Fields
Attributes Form
Joins
Auxiliary Storage
Actions
Display
Rendering
Temporal
Variables
Elevation
Single Symbol
Marker
Geometry Generator
Line
Arrow
Fill
Simple Fill
Symbol layer type
Simple Fill
Fill color
Fill style
Solid
Stroke color
Stroke width
0,200000
Millimeters
Stroke style
Solid Line
Join style
Bevel
Offset
x
0,000000
y
0,000000
Millimeters
Layer Rendering
Opacity
96,9 %
Blending mode
Layer
Normal
Feature
Normal
Draw effects
Control feature rendering order
Style
OK
Cancel
Apply
Help

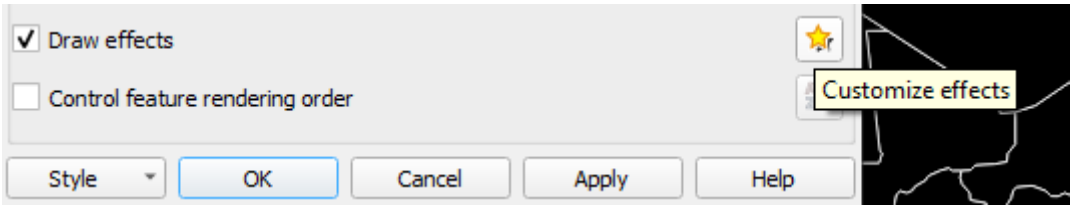
4. Visualisation basemap

In order to create a night-effect, a black background is recommended.

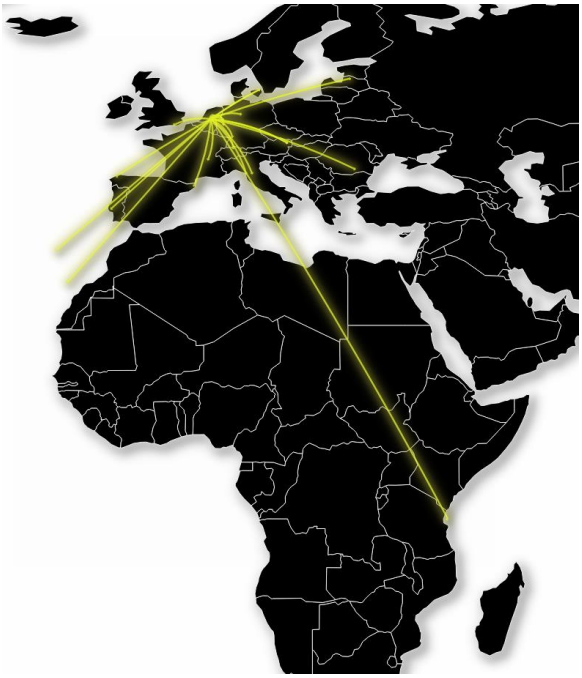
- Adjust the layer Countries to your liking.

The countries can pop-out a bit more by using draw effects.

- Open Layer Rendering in the lower part → select Draw Effects and click on the star



- Turn on Source, Outer Glow and Drop Shadow and adjust to your liking.
- Click on OK and Apply

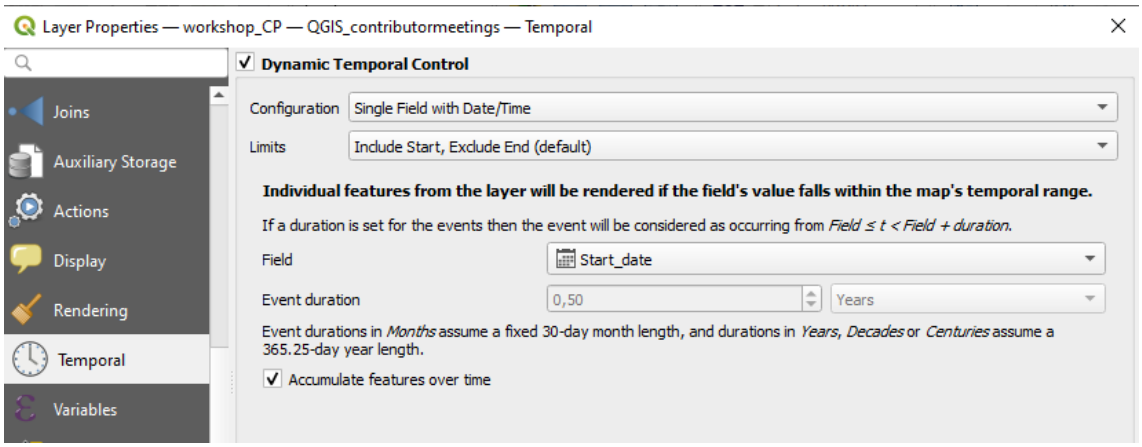


5. Temporal controller

Now it is time to visualize the evolution through time.
First we will add temporal settings to the layer of contributor meetings, before we set the temporal controller for a animation of the meeting location through time.

5.1 Turn on Temporal Settings of layer

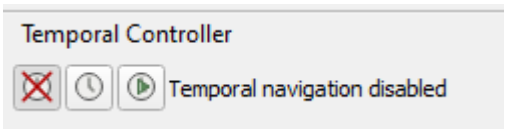
- Open the properties of the layer QGIS_contributormeetings
- Go to *Temporal* tab of the properties
- Turn on Dynamic Temporal Control
    - Set Configuration on *Single Field with Date/Time*
    - Set Field to *Start_date*
    - Turn on *Accumulate over Time*



5.2 Temporal Controller Panel

Temporal settings can be adjusted in the Temporal Controller Panel.

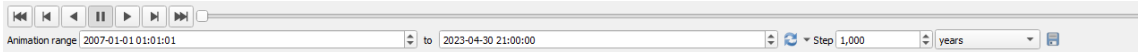
- Turn the *Temporal Controller Panel* on (by right-click on the empty space next to the toolbars or by using View → Panels)



- Enable *Temporal navigation*

Now it is time to set setting up the animation

- Set the Animation range from 2017 to end of April in 2023.
- Make the steps 1,000 month



- In the settings on the right , set the frame rate at 6 frames per second

Beautiful isn't it? Although, it is difficult to see in which cities the events are held.
Therefore we will add labels of the venues that will be visible for a short time.

5.3 Add timed labels

In order to add labels, that won't accumulate over time, we have to add an extra copy of the current QGIS_contributermeetings layer. With this copy we will set specific Temporal settings.

- Duplicate the layer QGIS_contributormeetings, call this layer QGIS_contributormeetings_labels
- In Properties, add a label on the value *Venue* in colors to your liking.
    - In this case we keep them black, with a yellow opaque buffer.
- In Properties, turn off symbology

Also for this layer we will turn on the temporal control on.
This time we will have other settings:

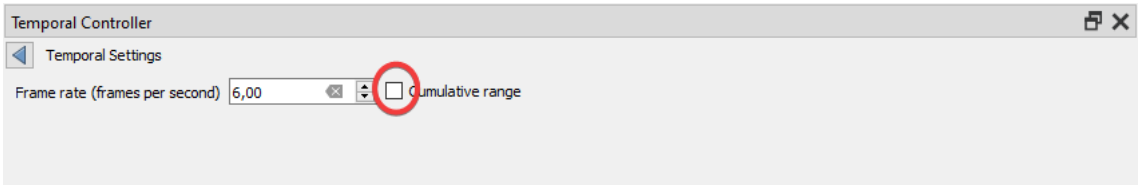
- In Properties, set the Configuration on Start and End Date / Time from Expressions
- Limits on *Include Start, Include End*
- Start Expression is *Start_date*
- End Expression is

```
make_date(year("Start_date")+1, month( "Start_date"), day( "Start_date"))
```

- Click on OK

Now turn on the *Temporal navigation* again.

- Also turn off 'Cumulative range' under the settings of the Temporal Controller:

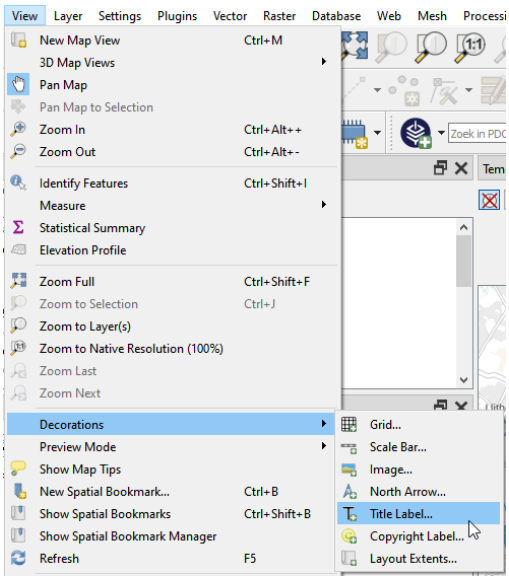


Beautiful isn't it? There is still an overlap, due to the fact that conferences are not exactly 12 months apart. The first person who makes an expression that creates an attribute 10 months after the start date, or has another clever solution receives a Groninger Koek (local treat from our home town)

6. Map decoration

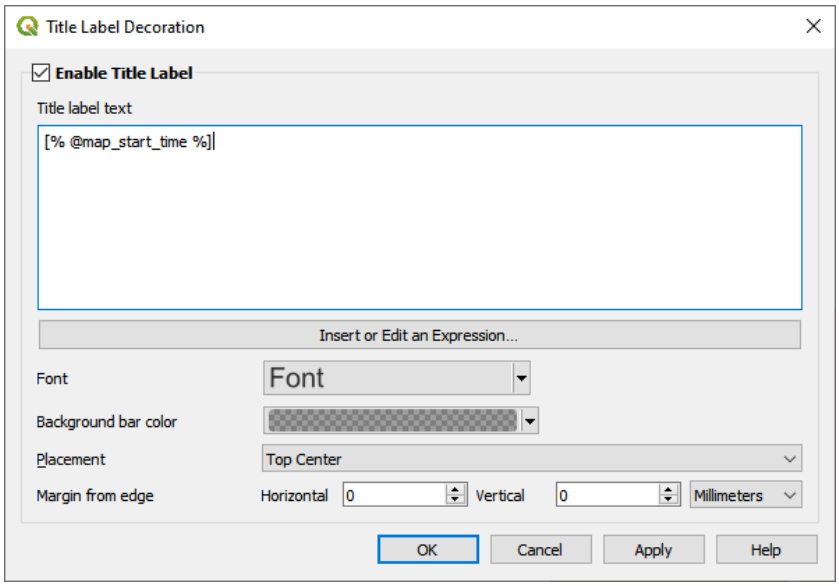
In this step, we're going to dynamically add the date and time of the current frame to our map, using a map decoration. This will show some extra information in our video.

- Add a Title Label or Copyright Label to your map: View > Decorations > Title Label or Copyright Label



- Add and expression to show the current map date:

```
@map_start_time
```



By default, it will look something like this:

Not great, could be better. Let's format the date.

- Format the date to something to your liking with the function **format_date**. For instance:

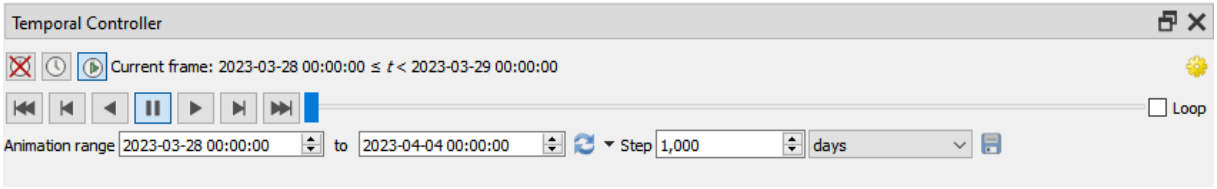
```
format_date( @map_start_time , 'MMM yyyy')
```

See the in-built documentation window on **format_date** on how to use it.

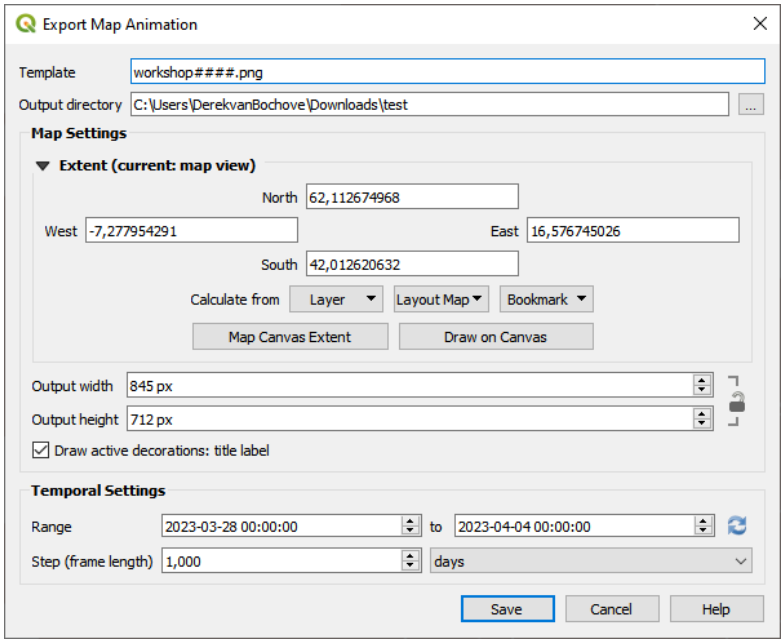
7. Export

Finally, we're going to export our map to frames (PNG images) that later on we can use to generate our movie.

- Make sure the settings in the Temporal Controller panel are OK for you (the animation range and the step)



- Click the save icon
- Choose an output location. The #### means: a sequence of 4 positions: 0001, 0002, etc.
- Make sure that what you want to show, is within the extent. The output width and height are linked to the extent. If you increase these, the extent will also change (experiment a little with the settings)



- Make sure the box next to "Draw active decorations" is ticked.
- Click "Save" and QGIS will export each map frame as a PNG image.

This could take a while… time for some chatter with your neighbour 😉

8. Make a movie out of the data

Now that we have the separate frames as PNGs, let's make a movie with them. For this, we'll use ffmpeg. This is a commandline application to convert between video and audio formats. But it can also be used to convert a sequence of images to video.

- Download ffpmeg for your OS at https://github.com/BtbN/FFmpeg-Builds/releases/
- Extract the zip somewhere on your machine
- Go to the folder where your png exports are located
- Open a console (commandline) in this folder
    - Or navigate in the commandline window to this folder.
- Run the following command (see below for explanation)

```
"path\to\ffmpeg.exe" -r 8 -f image2 -i frame%04d.png -vcodec libx264 -crf 15 -pix_fmt yuv420p output.mp4
```

In which:

- `"path\to\ffmpeg.exe"` should point to the location where you just extracted the zip
- `-r 8` means: 8 frames per second. A higher number of frames will result in a shorter video. Adjust this to your liking, or experiment a little with this setting.
- `-i frame%04d.png` should point to your frame PNGs
- `-crf 15:` sets the quality. 15 to 25 should be OK
- `output.mp4:` your output filename
- The rest of the settings: I'll leave this codec stuff to video nerds to explain, we're not qualified 😉

- Hit enter and magic will happen
- Show your results!