

## OmniSuite Story aus einem Video erstellen:

Software: ffmpeg, QGIS, PhotoShop, OmniSuite

Am Beispiel: OLLIE BYE: History Of The World: Every Year

- **Youtube Video in höchster Auflösung herunterladen**

> Qualitativ sind Konvertierungen in MP4 offensichtlich hochwertiger

> Mit PhotoShop lassen sich kein MP4-Videos öffnen

> <https://y2down.cc/de/> funktioniert aktuell (solche Services ändern sich ab und zu oder versuchen Premium-Angebote anzubieten): URL eingeben und Video in 1080p (FullHD) runterladen, falls das Video in der Auflösung angeboten wird

- **Video in PNG-Frames konvertieren mit ffmpeg**

○ unter <https://github.com/BtbN/FFmpeg-Builds/releases> das aktuelle Zip "ffmpeg-....-win64-gpl-shared.zip" laden und extrahieren

○ in dem entpackten Ordner unter "bin" die Datei ffmpeg.exe mit rechts anklicken und den Pfad kopieren

○ im Windows Startmenü die "Eingabeaufforderung" starten und einen Befehl aus kopierten Pfaden konstruieren (Strg+V zum Einfügen), nach folgendem Muster:

"ffmpeg.exe -i Videodatei.mp4 -vf fps=# Ausgabedatei.png"

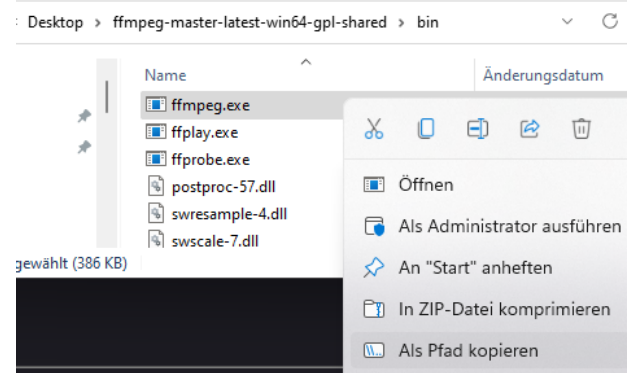
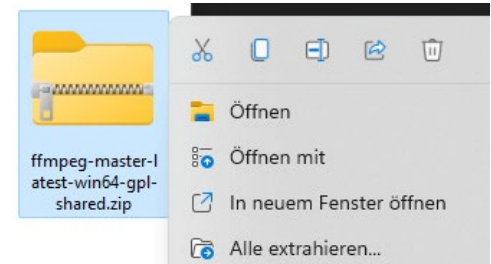
○ die exe muss mit vollem Pfad angegeben werden, die Videodatei ebenso und auch die Ausgabedatei

○ bei fps= wird eine Zahl angegeben wie z.B. 5, also fps=5, um pro Sekunde nur 5 Bilder zu extrahieren (YouTube-Videos haben meistens 30 frames per second)

- wenn zu viele Bilder übersprungen werden oder zu viele generiert werden, muss die Zahl niedriger (weniger Bilder) oder höher (mehr Bilder) gesetzt werden

○ eine komplette Befehlszeile kann wie folgt aussehen, wenn das Programm auf dem Desktop liegt (die Anführungszeichen nicht löschen):

- "C:\Users\Vigerske\Desktop\ffmpeg-master-latest-win64-gpl-shared\bin\ffmpeg.exe" -i "H:\Extract\The History of the World Every Year 1080p.mp4" -vf fps=5 "H:\Extract\bilder\%05d.png"
- das Video liegt bei mir im Ordner Extract auf der Festplatte H und die Bilder sollen in einen entsprechenden Unterordner (der muss vorher angelegt werden!)
- das %05d steht für eine Nummerierung mit 5 Stellen, das .png gibt das Dateiformat vor



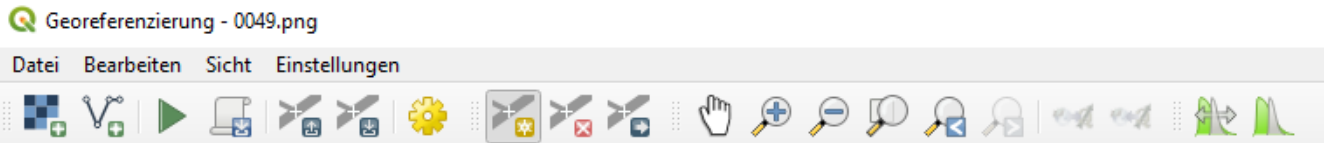
```

ca. Eingabeaufforderung
Microsoft Windows [Version 10.0.22000.2176]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Vigerske>"C:\Users\Vigerske\Desktop\ffmpeg-master-latest-win64-gpl-shared\bin\ffmpeg.exe" -i
"H:\Extract\The History of the World_Every Year 1080p.mp4" -vf fps=5 "H:\Extract\bilder\%05d.png"
  
```

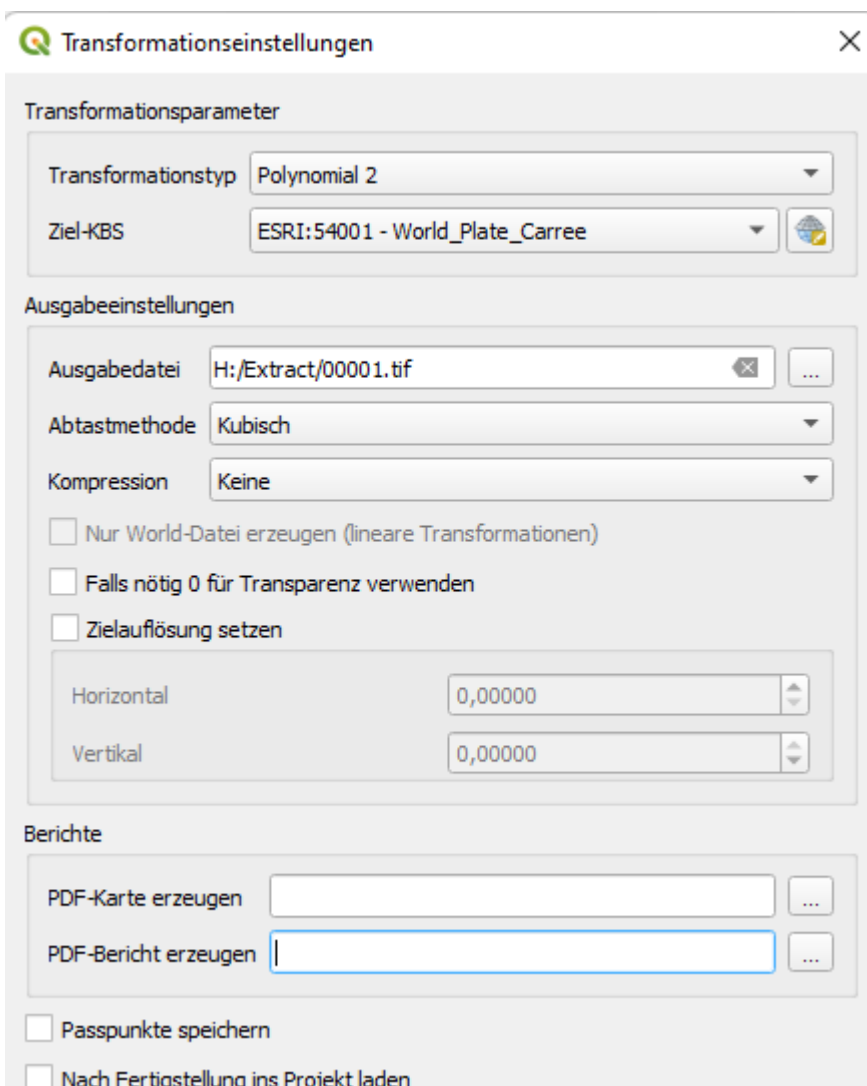
- **Georeferenzieren mit QGIS**

- in QGIS die Projektion des Videos einstellen (oder eine ähnliche, wenn nicht bekannt), in unserem Beispiel "Robinson Projection"
- in QGIS einen Weblayer oder eine andere Karte als Referenz einstellen, wenn die Karte keine Koordinaten zeigt



- Layer > Georeference

- Datei > eines der Bilder öffnen > Punkte setzen (Koordinaten eingeben oder Kartenreferenz nutzen)
- Einstellungen > Transformationseinstellungen (Typ meistens Polynominal order 2 (oder 1, wenn wenige Punkte ausreichen), Ausgabeort wird später noch angepasst, Abtastmethode mindestens kubisch)



- Datei: GDAL-Skript erzeugen

- das Skript würde EIN Bild georeferenzieren, wir wollen jedoch alle Bilder hintereinander bearbeiten und gleichzeitig umprojizieren und beschneiden

- **Skript anpassen, um zu georeferenzieren und umprojizieren**

- das Skript aus QGIS z.B. in den Editor einfügen und als georeferenzieren.bat (nicht txt) in den Ordner speichern, der den Unterordner mit den Einzelbildern enthält, bei uns H:/Extract/
- es enthält zwei Befehle: gdal\_translate setzt die Referenzpunkte für die Georeferenzierung und speichert das Ergebnis in einen Temp-Ordner
- gdalwarp enthält die Transformationsparameter, ruft das Bild aus dem Temp-Ordner auf und projiziert es in die angegebene Projektion
- wir fügen dem Skript jetzt eine Schleife hinzu, die alle Bilder im Ordner bearbeitet und in einen neuen speichert
- dabei ändern wir das Ziel-Koordinatensystem und beschneiden das Ergebnis
- unser Code sieht zunächst so aus

```
gdal_translate -of GTiff -gcp 148.811 201.396 -1.06076e+07 6.36598e+06 -gcp 1651.98 191.887
1.16695e+07 6.52119e+06 -gcp 512.377 1022.43 -5.15571e+06 -5.80444e+06 -gcp 1701.91 924.623
1.25322e+07 -4.38174e+06 -gcp 817.443 328.415 -563699 4.48154e+06 -gcp 957.623 833.604 1.55736e+06 -
2.99845e+06 -gcp 828.651 76.5 -670517 8.2053e+06 -gcp 286.953 527.009 -8.36775e+06 1.54477e+06 -gcp
1749.54 649.783 1.33145e+07 -279695 "H:/Extract/bilder/3329.png"
"C:/Users/Vigerske/AppData/Local/Temp/3329.png"
gdalwarp -r cubic -order 2 -co COMPRESS=None -t_srs "+proj=eqc +lat_ts=0 +lat_0=0
+lon_0=0 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs"
"C:/Users/Vigerske/AppData/Local/Temp/3329.png" "H:/Extract/3329.png"
```

und enthält die **Georeferenzierungspunkte**, eines der **Input-Bilder**, die **temporäre Ausgabe**, die **Transformationsparameter**, die **Projektion** und **den Output**.

Als erstes fügen wir die **Schleife** ein, die dafür sorgt, dass alle Bilder prozessiert werden.

Eine Zeile vor gdal\_translate kommt folgender Code:

```
@echo off
for %%f in (bilder\*.png) do (
```

und hinter die gdalwarp-Zeile schließen wir die Klammer:

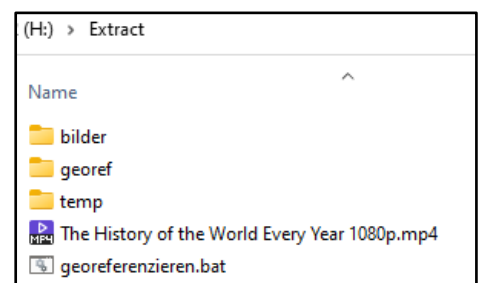
```
)
```

Jetzt werden alle png-Dateien im Unterordner bilder bearbeitet (vom Ort der bat-Datei ausgehend).

Und als Dateiname dient die Variable %%f. Diese setzen wir jetzt, leicht modifiziert (%%~nf), an mehreren Stellen ein (wir kürzen hier für die Ansicht die Punktkoordinaten raus):

```
@echo off
for %%f in (bilder\*.png) do (
    gdal_translate -of GTiff -gcp [...] "bilder\%%~nf.png" "temp\%%~nf.tif"
    gdalwarp -r cubic -order 2 -co COMPRESS=None -t_srs "+proj=eqc +lat_ts=0
+lat_0=0 +lon_0=0 +x_0=0 +y_0=0 +datum=WGS84 +units=m +no_defs"
    "temp\%%~nf.tif" "georef\%%~nf.png"
)
```

Zuerst wird der Pfad für die **Input-Bilder** neu gesetzt, dann **der temp-Ordner** und dann **der Output**. Alle drei Ordner müssen sich im gleichen Ordner wie die bat-Datei befinden.



Als nächstes ändern wir die **Projektion**. Dieser Schritt wird nur benötigt, wenn das Ausgangsmaterial nicht bereits in Plate Carree projiziert vorliegt, der Projektion, die die Globus-Software nutzt.

In unserem Beispiel steht bei gdalwarp, vor dem temp-Ordner, noch die komplette Definition der Robinson-Projektion. Das -t\_srs steht für target spatial reference system. Tatsächlich ist in unserem Beispiel nur die Quelle Robinson, als Ziel für den Globus benötigen wir Plate Carree. Dazu nutzen wir zusätzlich die Angabe -s\_srs (source). Außerdem können Projektionsangaben auch deutlich kürzer über den EPSG oder ESRI-Code angegeben werden, wie er in QGIS überall zu finden ist, wir ersetzen daher die lange Definition:

```
gdalwarp -r cubic -order 2 -co COMPRESS=None -s_srs ESRI:54030 -t_srs ESRI:54001
    "temp\%%~nf.tif" "georef\%%~nf.png"
```

Als Quell-Bezugssystem haben wir ESRI:54030 (Robinson) angegeben und als Ziel 54001 (Plate Carree).

Jetzt fehlen noch die Beschneidung auf die **Maße der Erde** sowie die Pixelmaße des finalen Bildes:

```
gdalwarp -r cubic -order 2 -co COMPRESS=None -s_srs ESRI:54030 -t_srs ESRI:54001
    -te_srs EPSG:4326 -te -180 -90 180 90 -ts 4096 2048
    "temp\%%~nf.tif" "georef\%%~nf.png"
```

Eine weitere Projektionsangabe wird nötig. Mit -te\_srs setzen wir die Projektion der Funktion -te auf WGS84 (4326). Als target extent (te) geben wir das Maximum der Erde an, Dank WGS84 in Grad.

Target Size (ts) setzt die Pixelmaße des finalen Bildes auf die, die der Globus typischerweise nutzt.

### Unser vollständiger Code sieht wie folgt aus:

```
@echo off
for %%f in (bilder\*.png) do (
    gdal_translate -of GTiff -gcp 148.811 201.396 -1.06076e+07 6.36598e+06 -gcp 1651.98 191.887 1.16695e+07
6.52119e+06 -gcp 512.377 1022.43 -5.15571e+06 -5.80444e+06 -gcp 1701.91 924.623 1.25322e+07 -4.38174e+06 -gcp
817.443 328.415 -563699 4.48154e+06 -gcp 957.623 833.604 1.55736e+06 -2.99845e+06 -gcp 828.651 76.5 -670517
8.2053e+06 -gcp 286.953 527.009 -8.36775e+06 1.54477e+06 -gcp 1749.54 649.783 1.33145e+07 -279695
"bilder\%%~nf.png" "temp\%%~nf.tif"
    gdalwarp -r cubic -order 2 -co COMPRESS=None -s_srs ESRI:54030 -t_srs ESRI:54001 -te_srs EPSG:4326 -te -
180 -90 180 90 -ts 4096 2048 "temp\%%~nf.tif" "georef\%%~nf.png"
)
```

- **Skript ausführen**

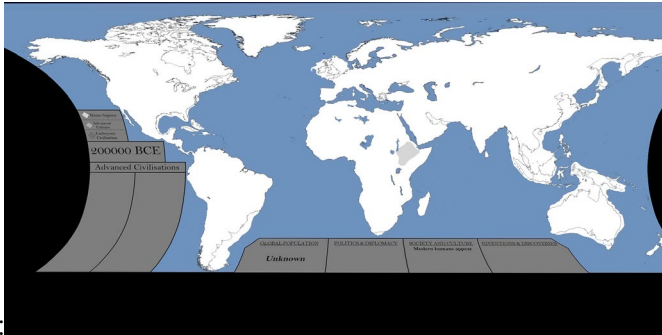
- mit QGIS wurde das Programm OSGeo4W Shell installiert

- darin auf das Laufwerk, in den Ordner mit der bat-Datei navigieren und ausführen, in unserem Fall:

```
> h:
> cd Extract
> georeferenzieren.bat
```

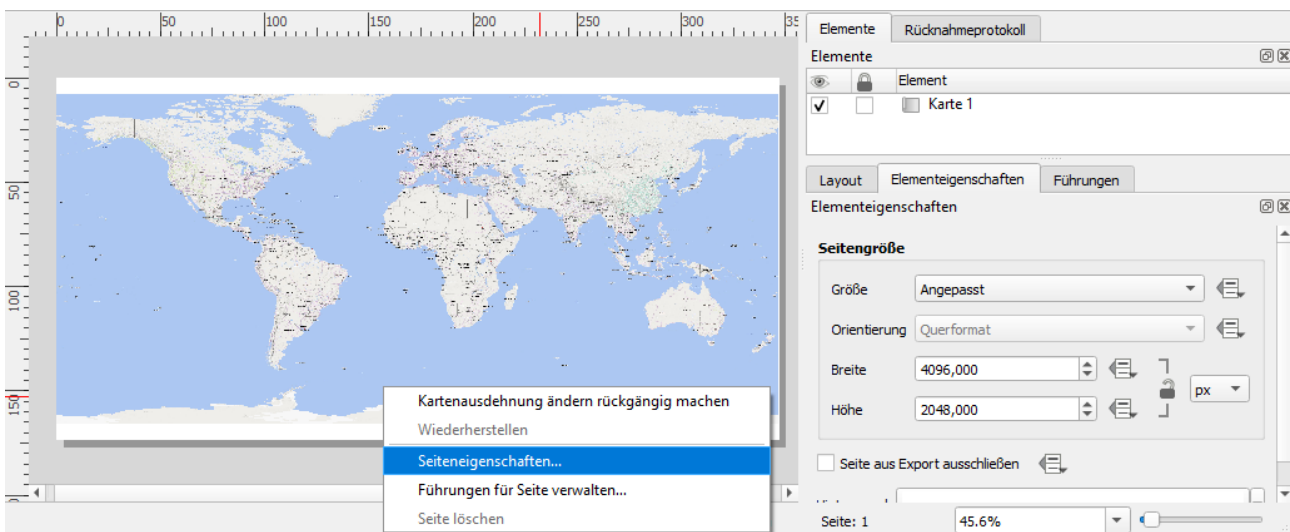
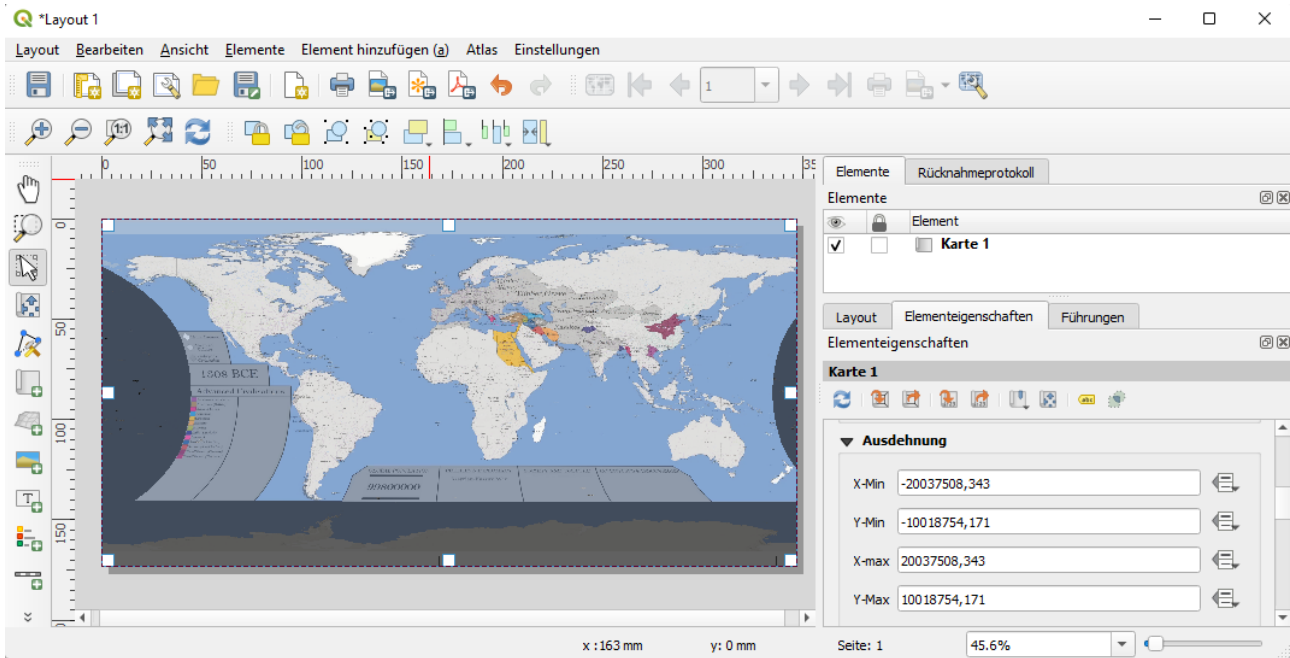
Es sollten alle png-Dateien im Ordner "bilder" georeferenziert, kurz unter "temp" zwischengespeichert, dann umprojiziert und beschnitten werden. Das Ergebnis landet unter georef. Am besten erstmal mit nur wenigen Bildern testen. Sollte die Karte stark verzerrt wirken, sollten evtl. einfach mehr Punkte gesetzt werden oder prüfen, ob die Angabe der Quell- und Ziel-Koordinatensysteme stimmen.

- **Aufbereitung der Bilder**



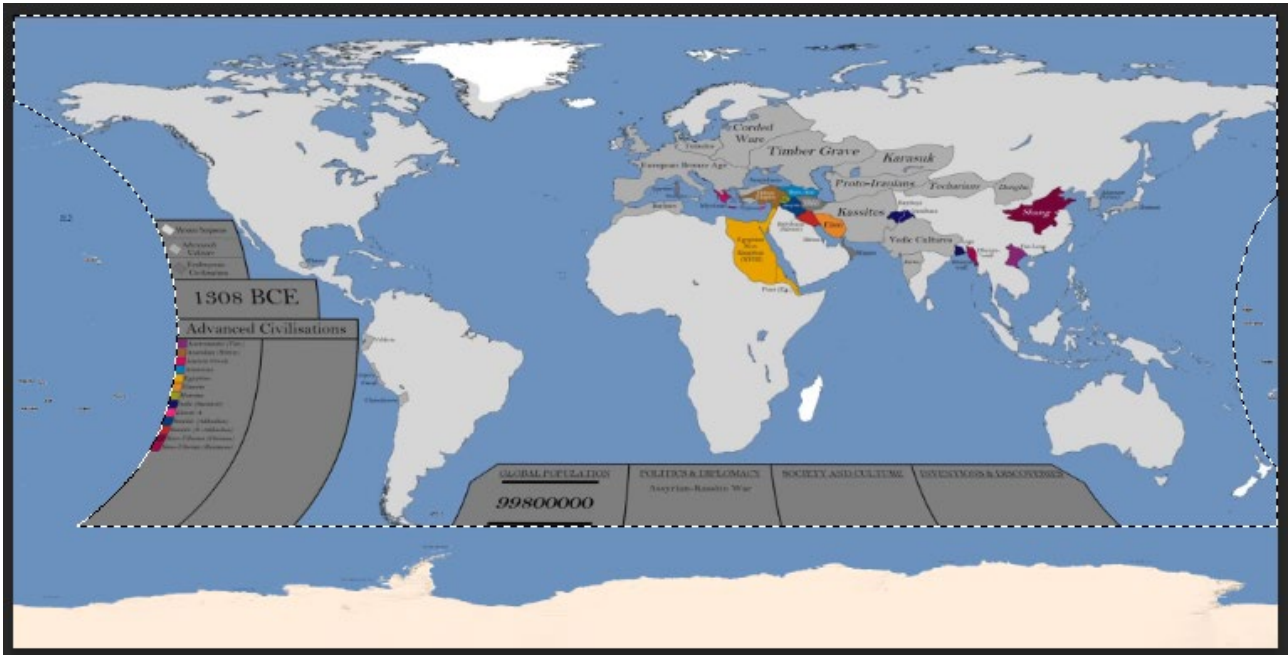
> Ausgangsbild:

- > in QGIS Layout mit Basemap erstellen, um die fehlenden Elemente (Nordpol, Südpol) zu ergänzen
- Kartenausdehnung fix definieren und in passendem Format (4096 x 2048) speichern

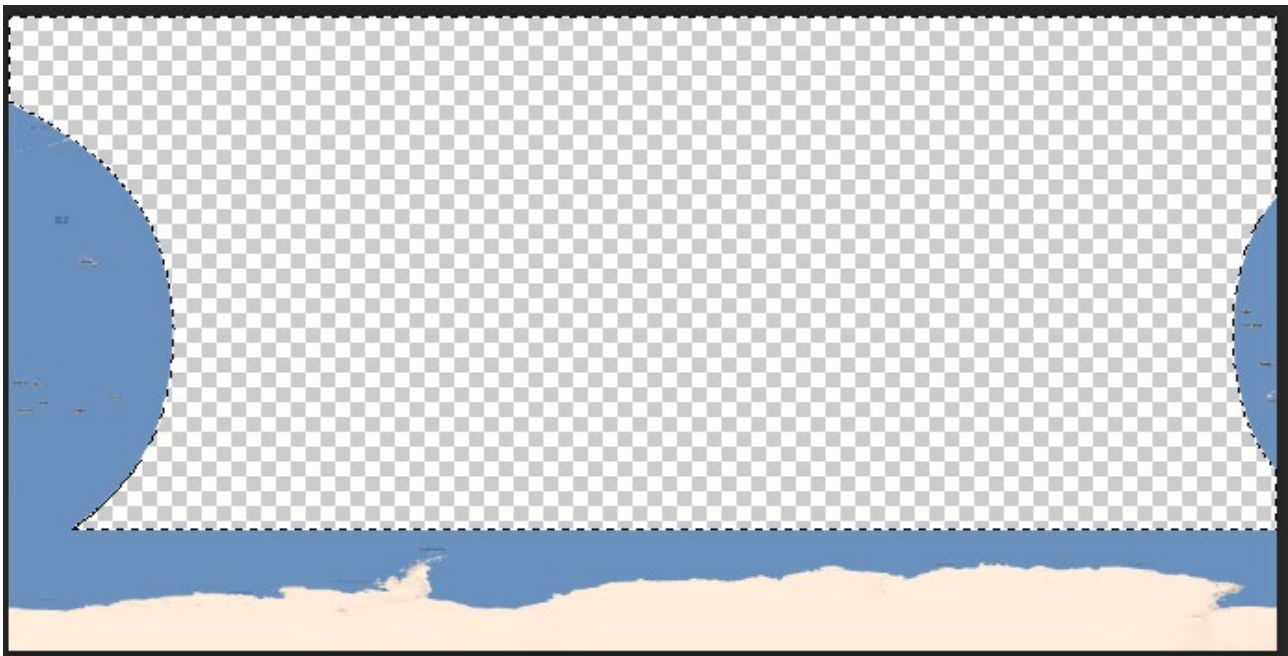


Die exportierte Karte wird als Maske im Globus-Material genutzt, um die Video-Karte zu ergänzen.

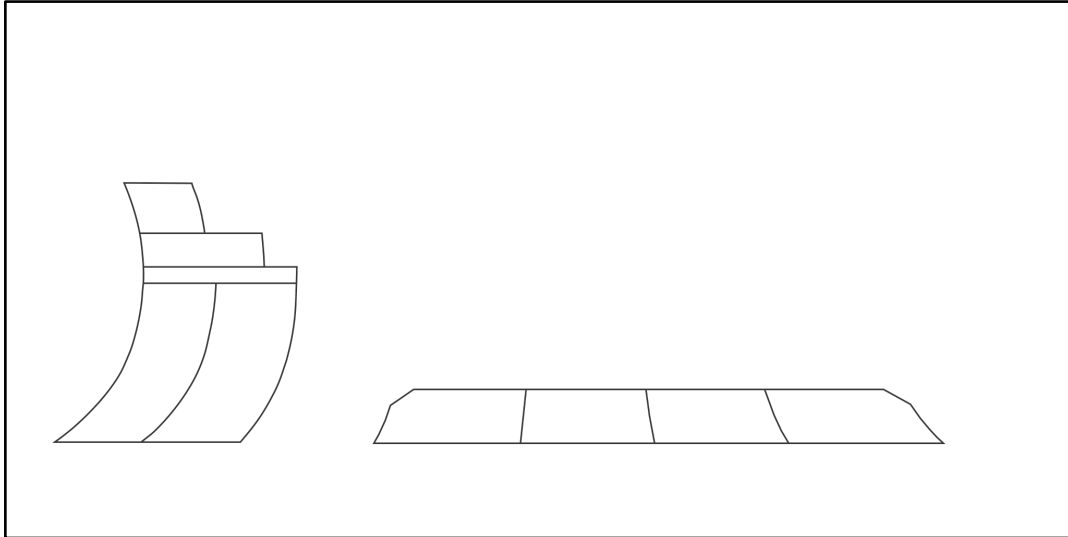
- > Georeferenzierte Karte aus dem Video in Photoshop laden und in eine Ebene darunter die Basemap aus QGIS
- Ränder in der Video-Karte entfernen bis die QGIS-Basemap nahtlos darunter zu sehen ist
- die QGIS-Basemap farblich an die aus dem Video anpassen
- den übrig gebliebenen Bereich der Video-Karte selektieren



- und diesen Bereich in der Basemap löschen, Video-Karte wieder ausblenden und das Ergebnis als PNG mit Transparenz speichern - wir haben eine Maske, die wir als zweite Textur über die Video-Karte legen können



> an der Stelle können auch noch weitere Masken erstellt werden, z.B. eine neue Kontur um die Legende:



- **Einladen in den OmniSuite Material-Editor**

- zuerst den Ordner mit den georeferenzierten Bildern als Animation
- danach die Maske mit der QGIS-Basemap
- bei Bedarf weitere Masken