

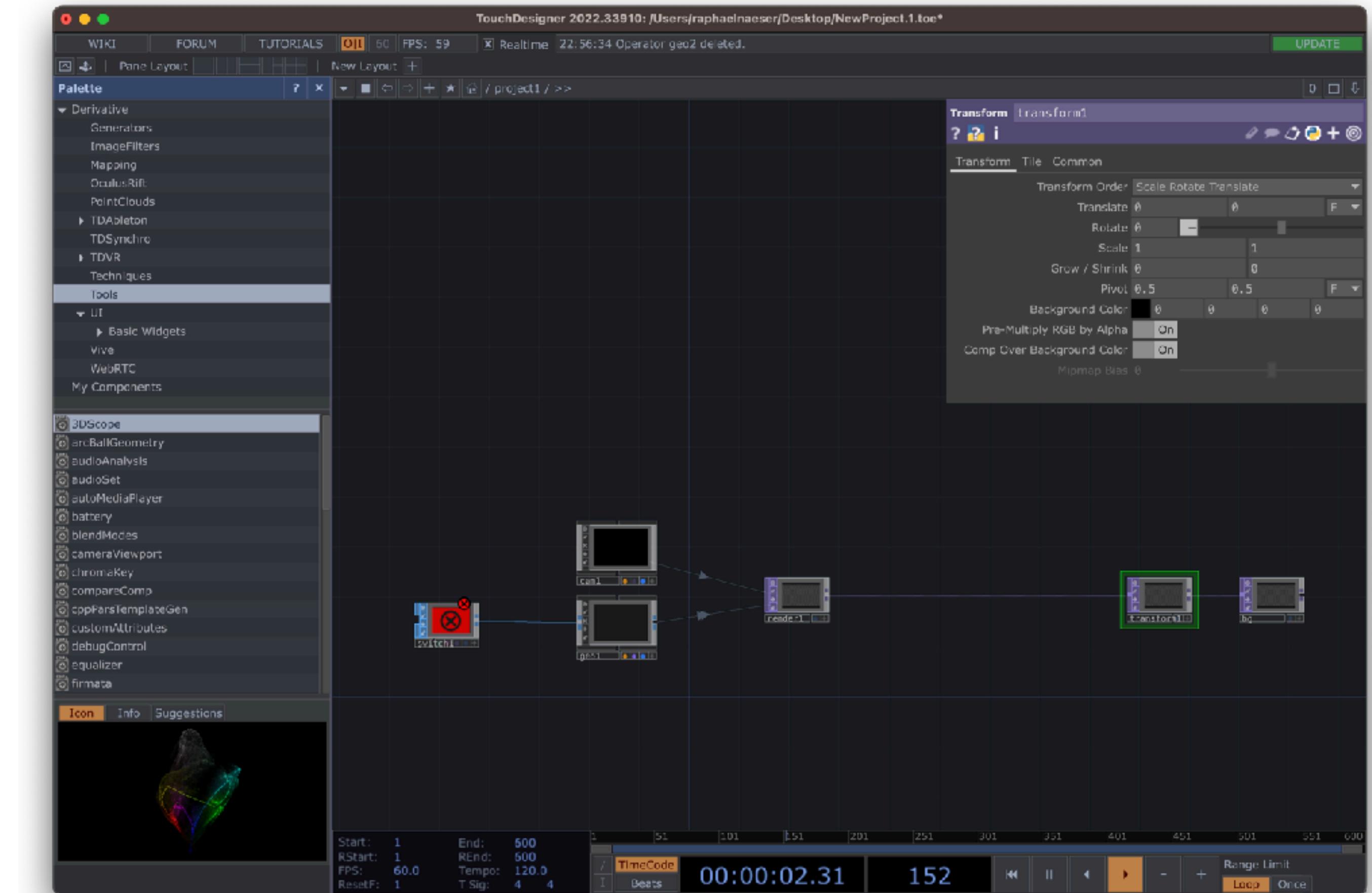
# **TD-Beginner 101 Workshop**

**v1.1**

**Drake Rubicon - 2024.07.17**

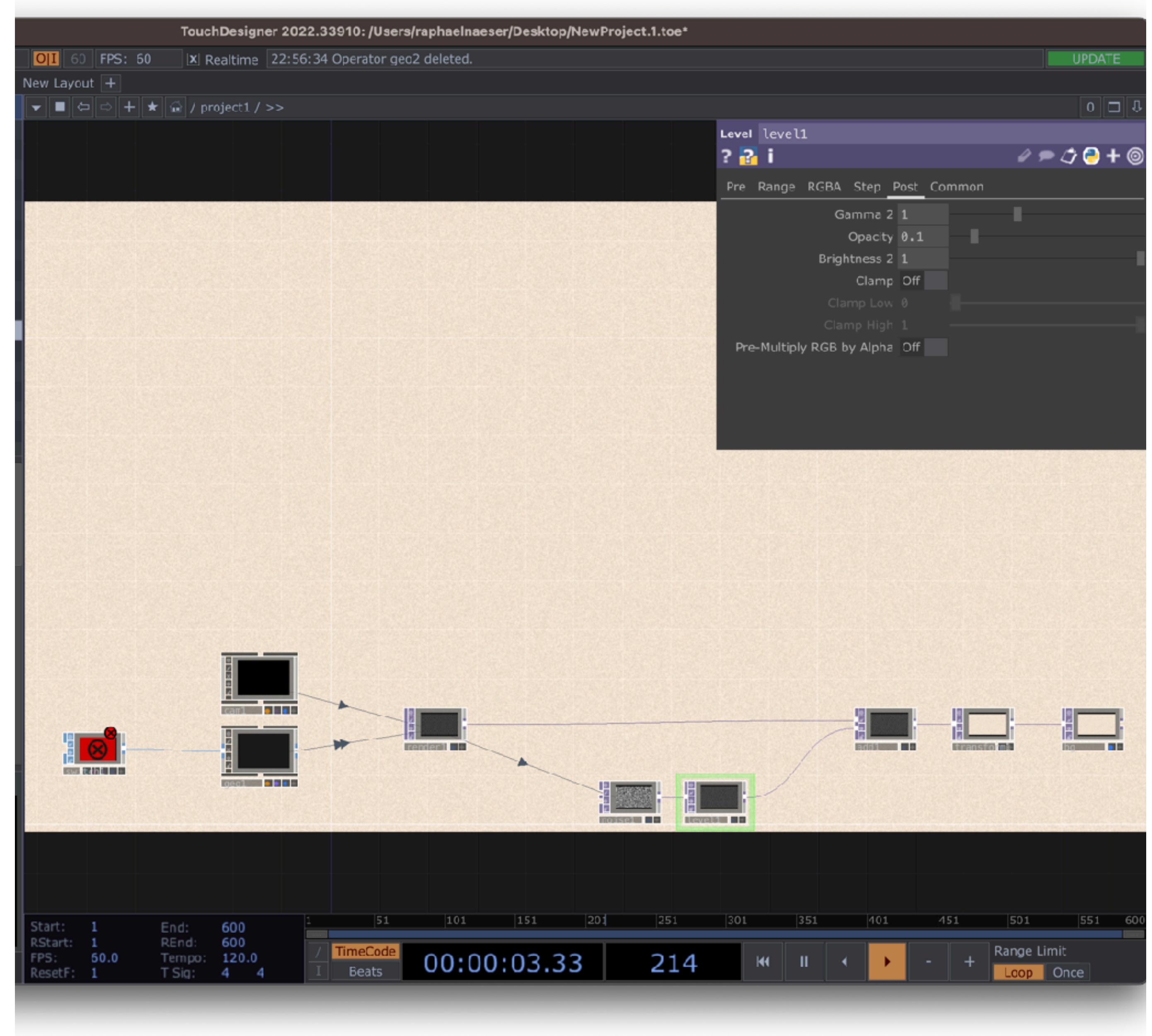
# **Part 1 - Handdrawn Lines**

- Basic Render Setup
  - Cam, Geo, Render
  - Geo -> Standardkörper -> In/Out
  - Autogeo von Switch Sop
  - Transform -> Comp over BG -> Hintergrund zu Papierfarbe
  - Null -> “out1”
  - Blauer Viewer-Punkt

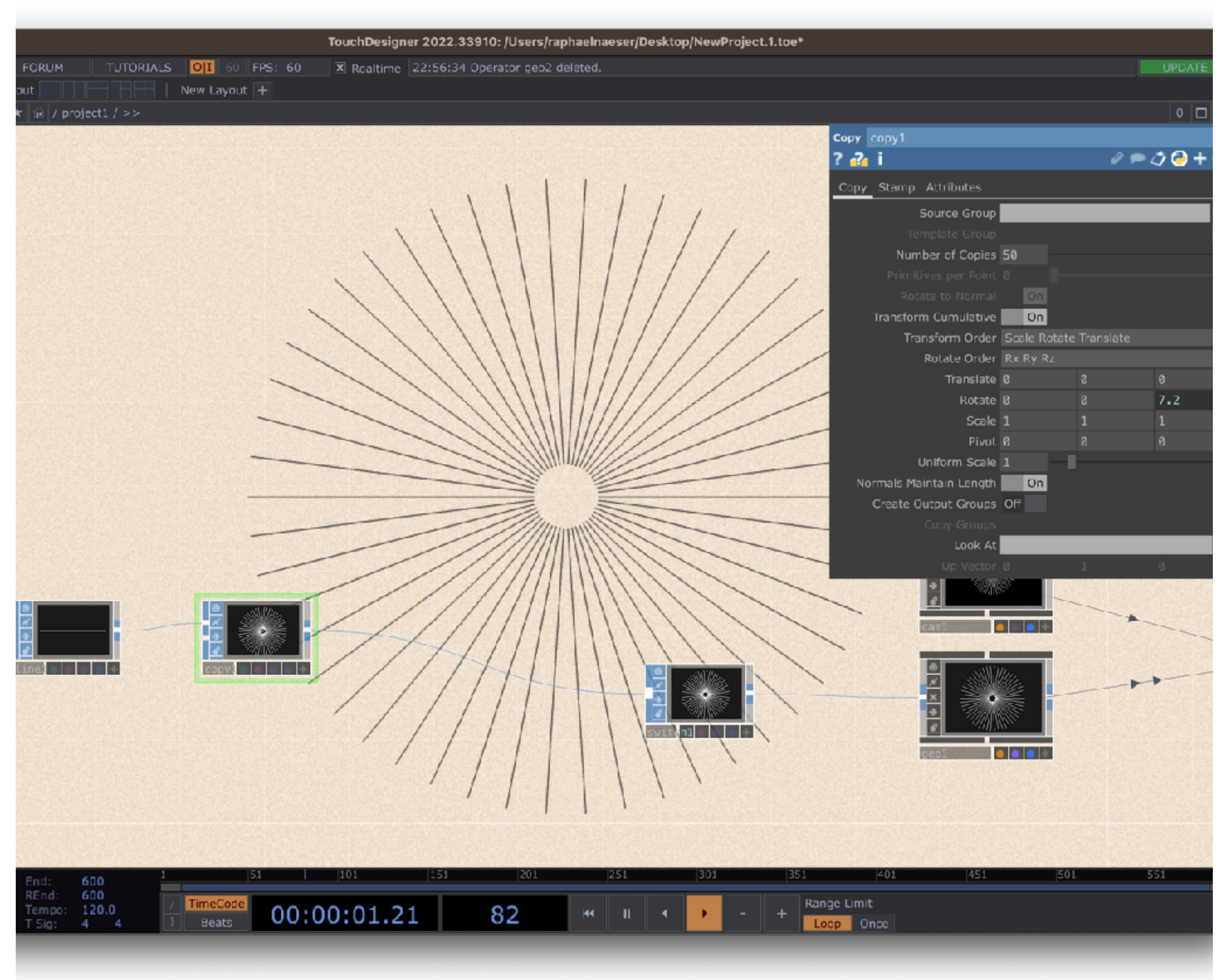


COMP - Components / TOP - Texture Operator

- Add auf Connection
- TOP Noise + Resolution:
  - op('render1').width
  - op('render1').height
  - Noise type random
- TOP Level
  - Post - Opacity 0.1

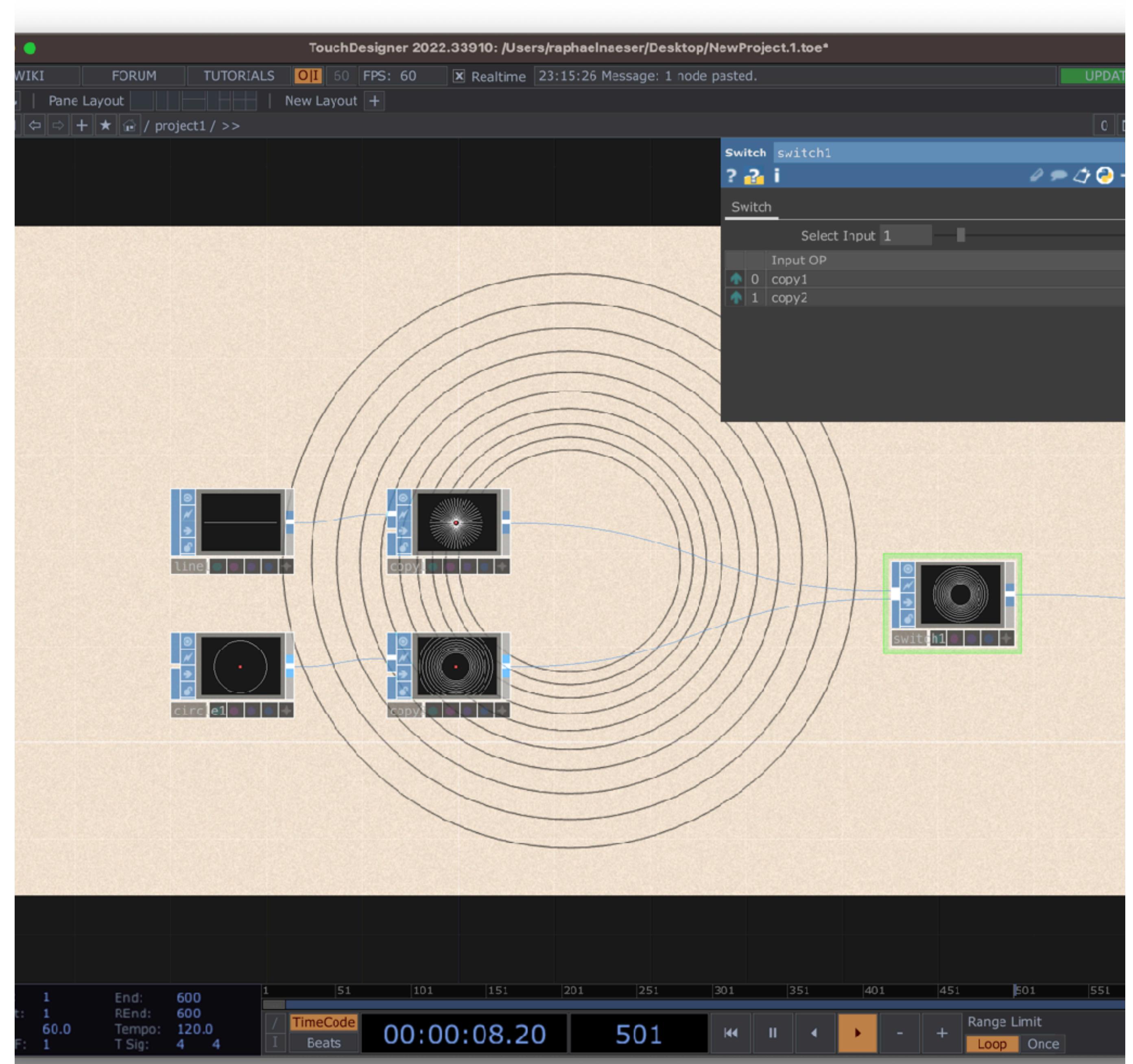


- SOP Line
  - Etwas verkürzen: Pa: .1 0 0
- SOP Copy
  - num of copies: 50
- Copy - Switch
- Ausrichten durch rotieren:
  - Rotate z:  $360/\text{me.par.ncy}$
  - Hinweis: Hover = Parametername



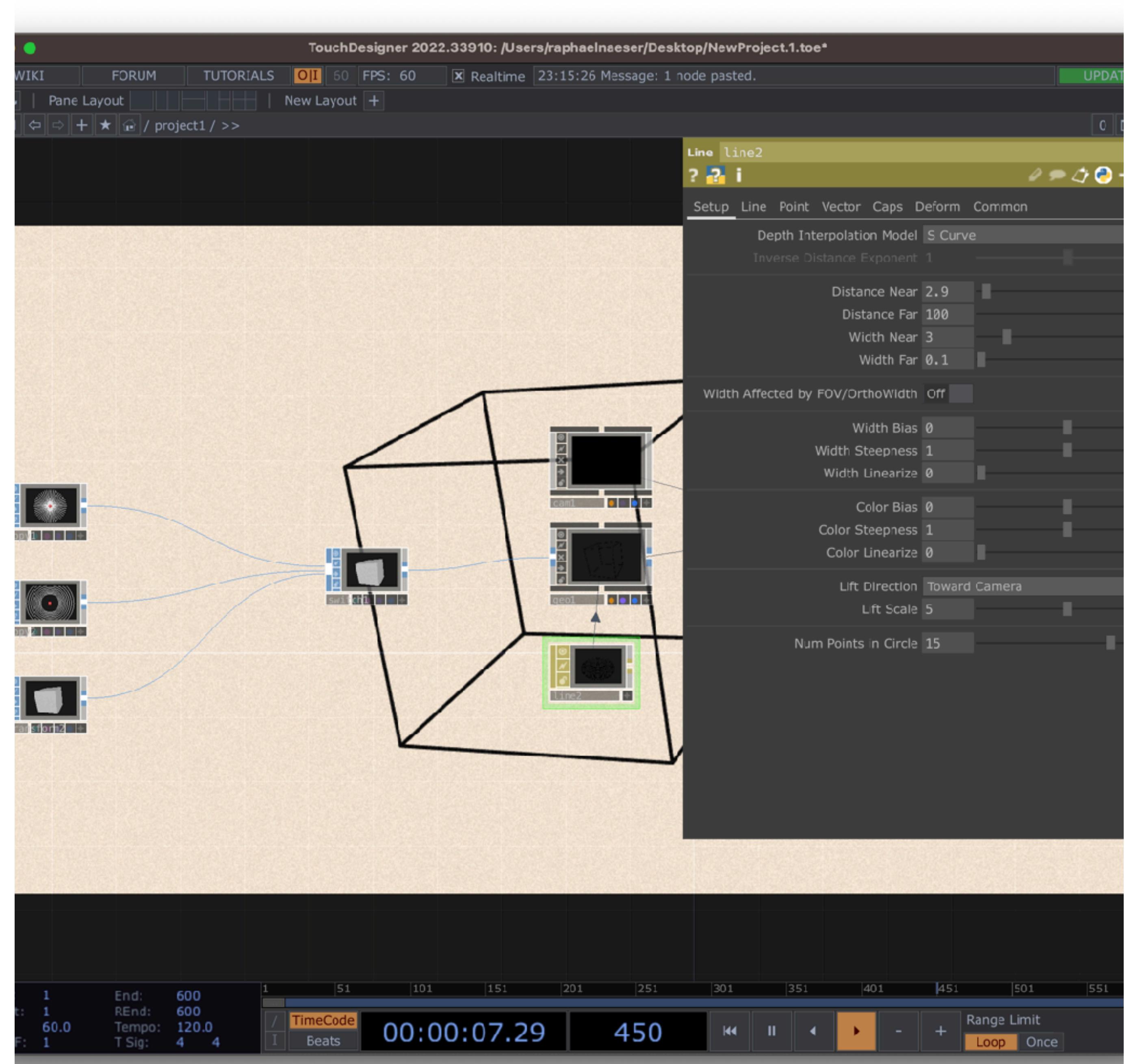
SOP - Surface Operator

- SOP Circle
  - Divisions: 100
  - Arc-Type: Open Arc
- SOP Copy
  - Uniform Scale 10
  - Copy - Switch



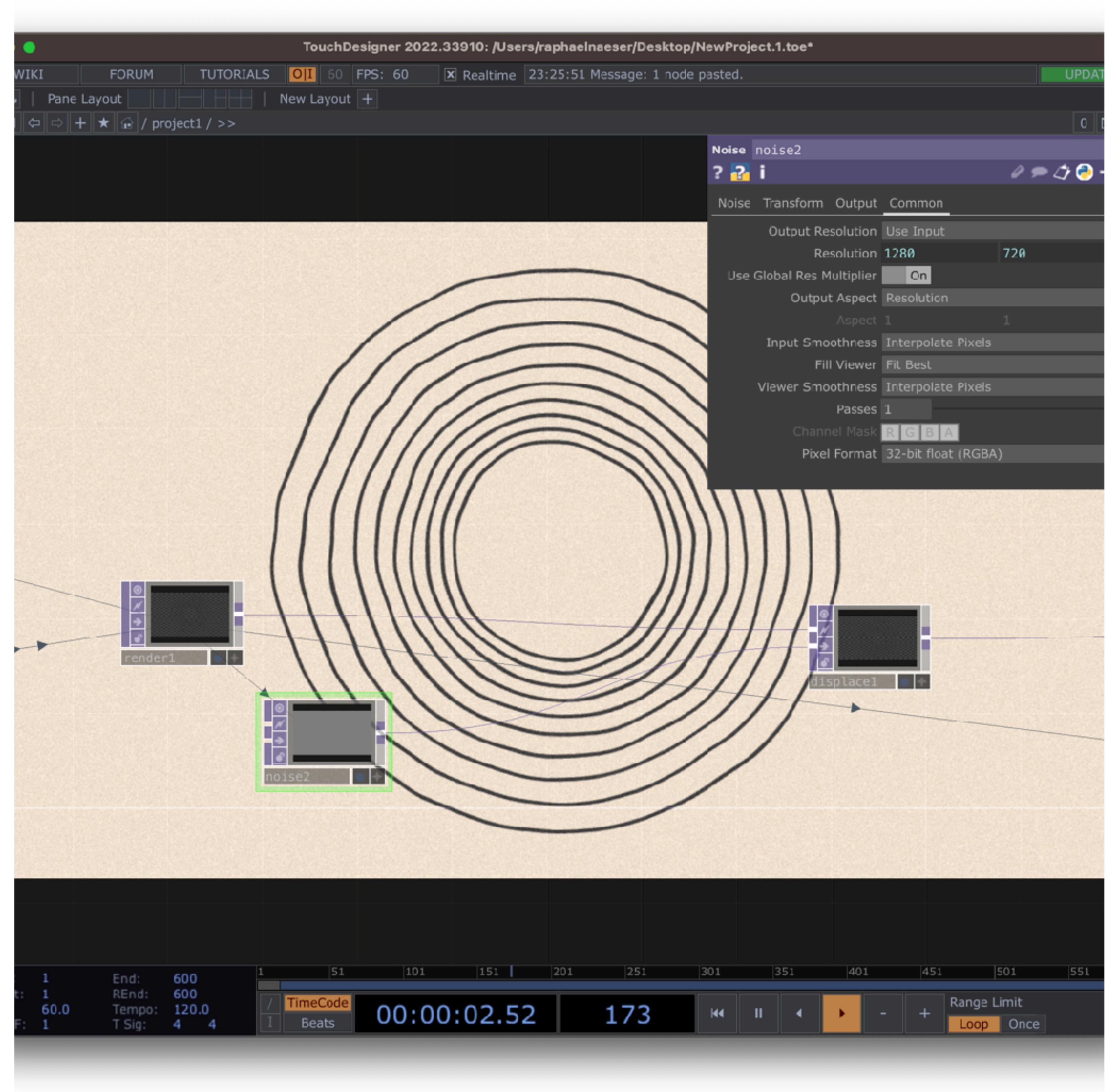
SOP - Surface Operator

- SOP Box
- SOP Transform
  - Rotation: 20 -25 0
- Vorschau im Switch: Volle Box
- 3D braucht Textur oder Material
- MAT Line
  - Auf Geo ziehen - Param Mat
  - Farbe wechseln Near/Far
  - Gleichm. Linien:
  - Dist. Far 100 // Width Near - Stärke



SOP - Surface Operator / MAT - Material

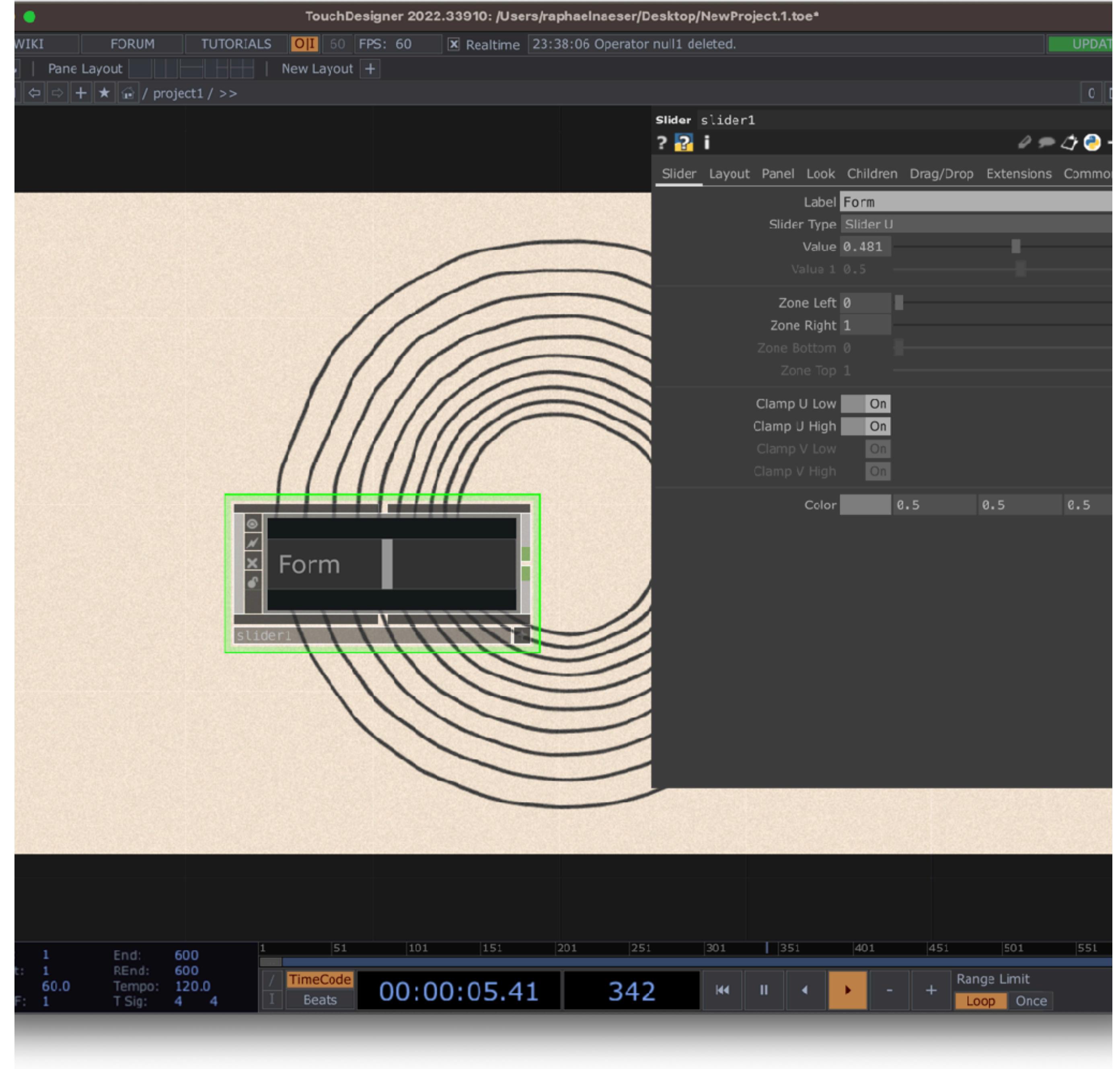
- TOP Displace
- Noise1 CopyPaste
- Verbinden
- Noise Type: Simplex 3D
- Monocrome: off
  - Amplitude 0.002
- Glitches beseitigen:  
Common  
Pixel Format 32bit float



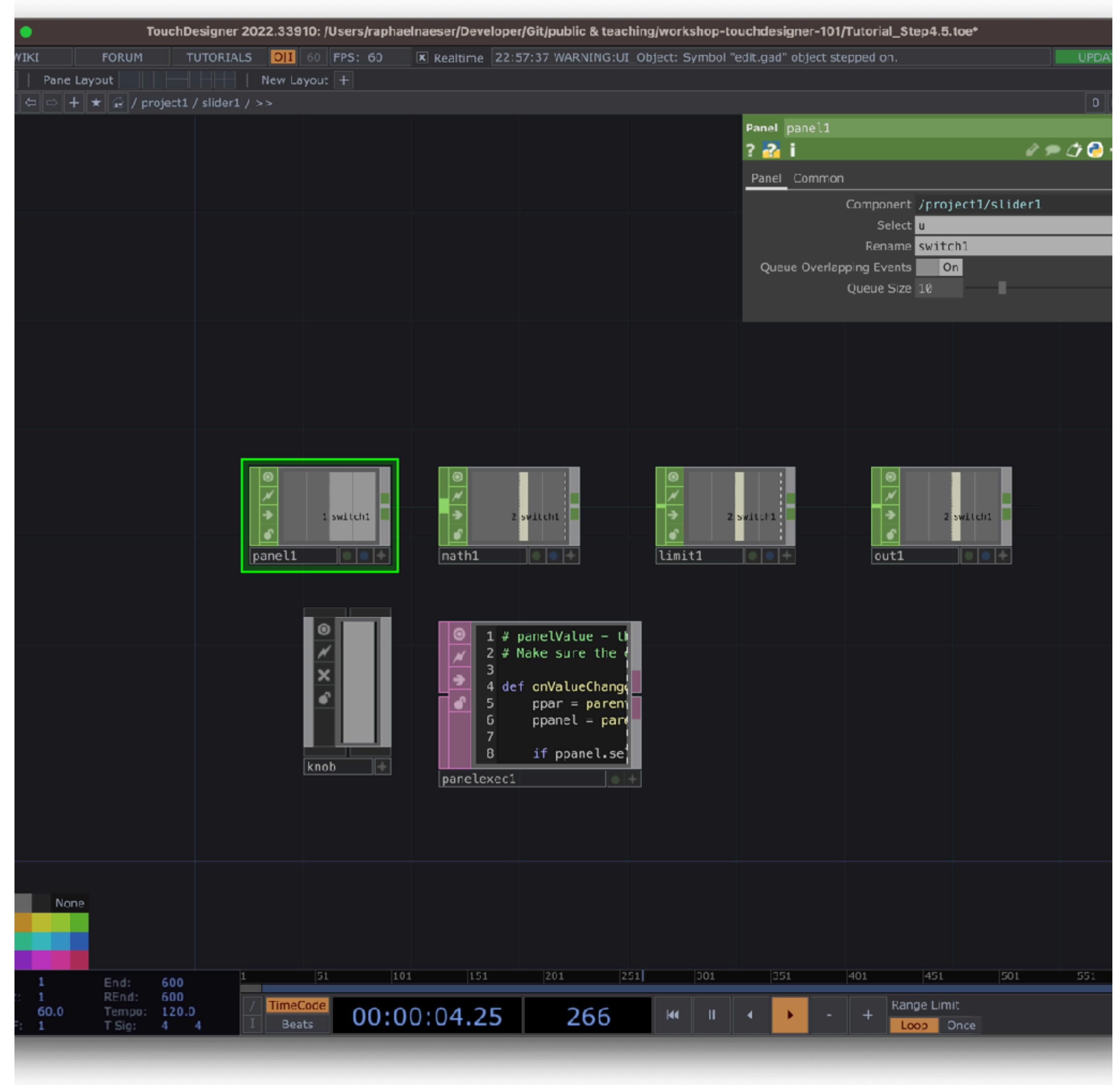
# **Part 2 - Interaktivität 1**

- COMP Slider

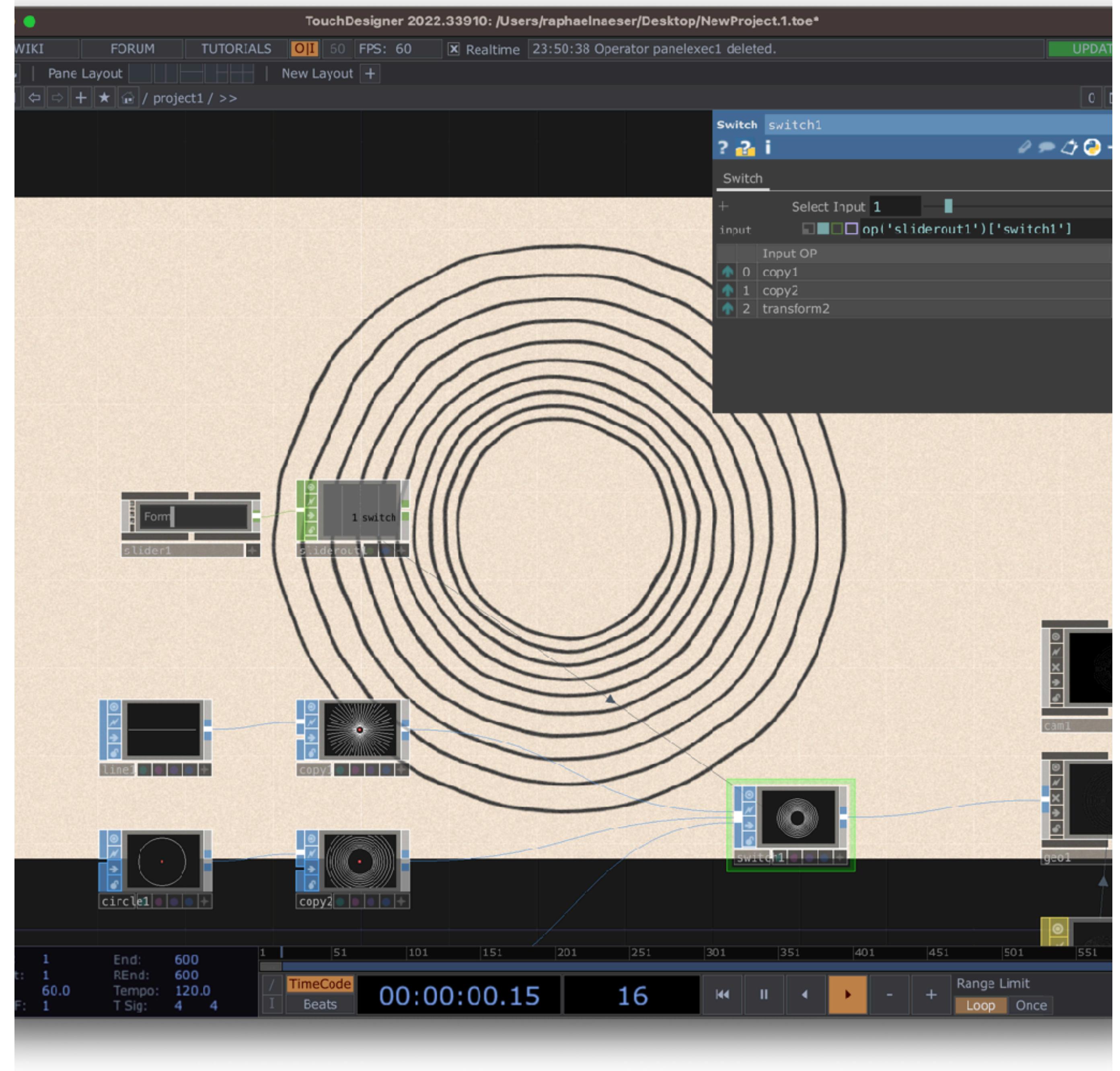
- Look: Farbe ändern
- Name Geben
- Autogeo von Switch Sop
- Viewer active -> Testen
- Slider geht von 0 bis 1
- 3 Formen -> 0, 1, 2
- Zoom in Slider



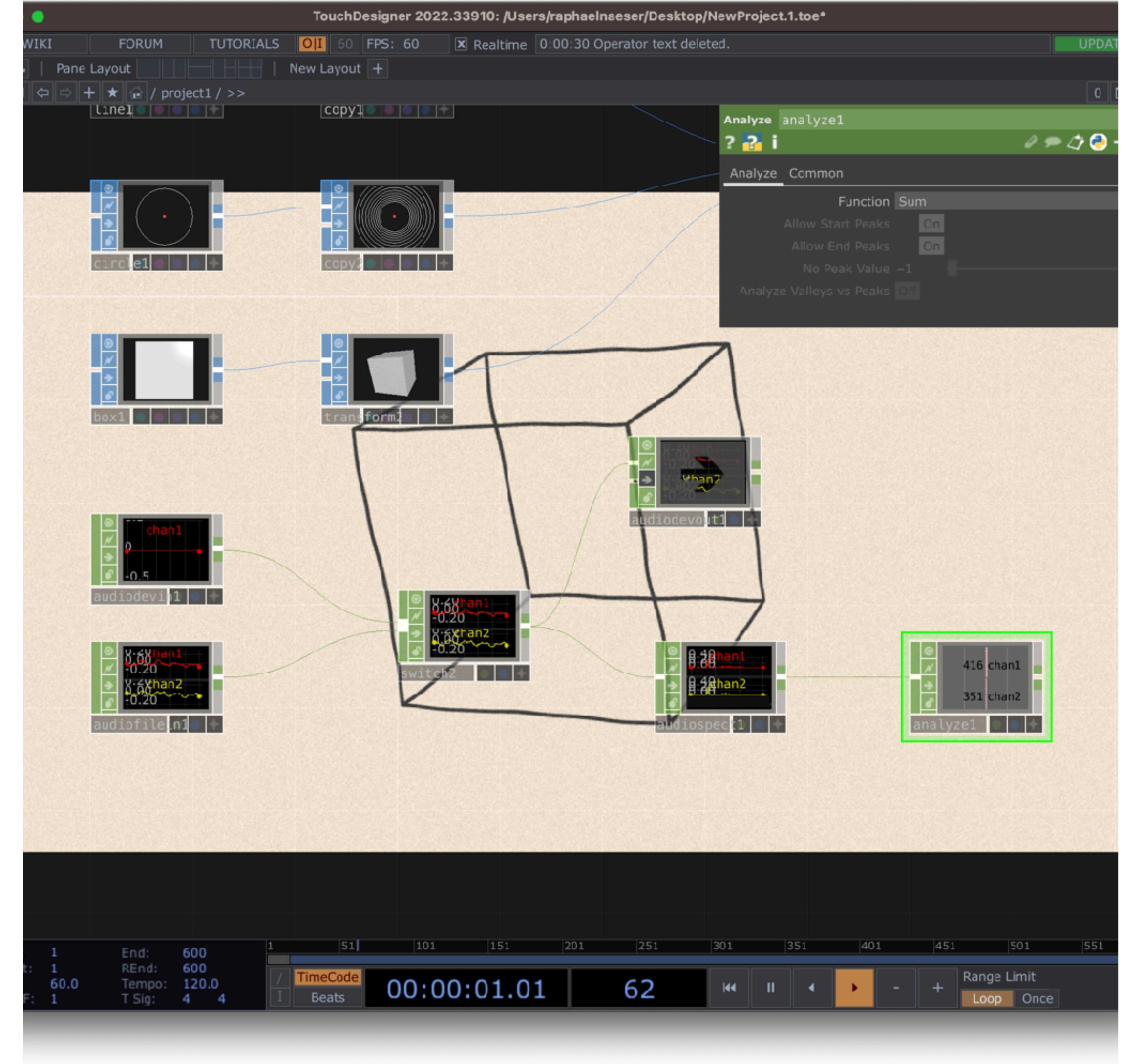
- par1 löschen
- CHOP panel1
  - select - u
  - Rename - switch1
- Out1 platz geben
- CHOP math1
  - Multiply 2
- CHOP Limit1
- Quantize Round
  - Step 1



- Rauszoomen
- CHOP Null
  - Umbenennen “sliderout1”
- SOP switch 1
  - Select Input
  - op('sliderout1')['switch1']
- Slider1 XY: 10 / 10
- Perform Window zeigen

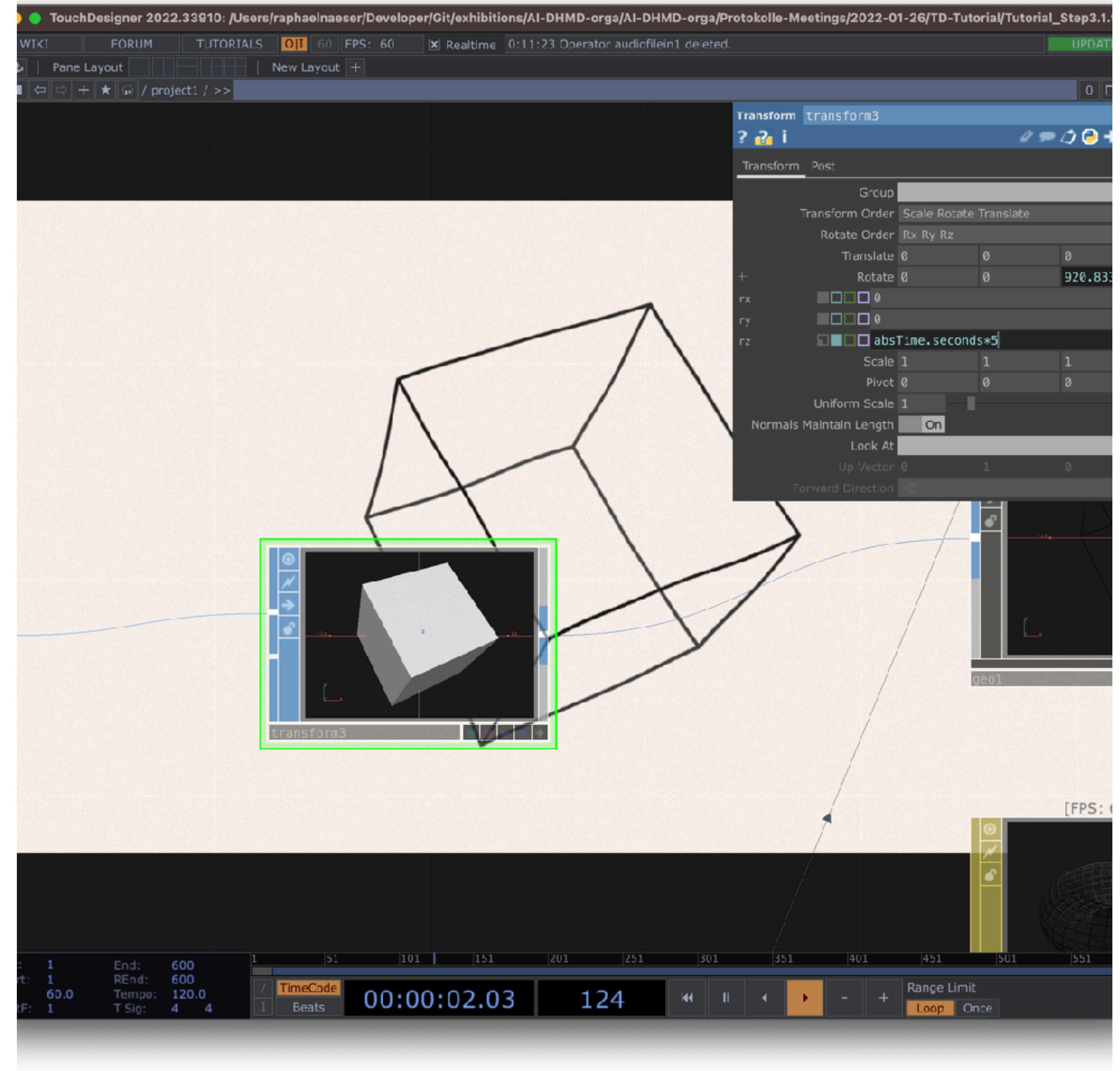


- CHOP Audiodevicein
- CHOP Audiofile
- CHOP Switch
- CHOP AudiodeviceOut
  - Bypass =>
- CHOP Audiodevicein
- CHOP AudioSpectrum
  - FFT 16384
- CHOP Analyze
  - Quantisiert



# **Part 3 - Interaktivität 2**

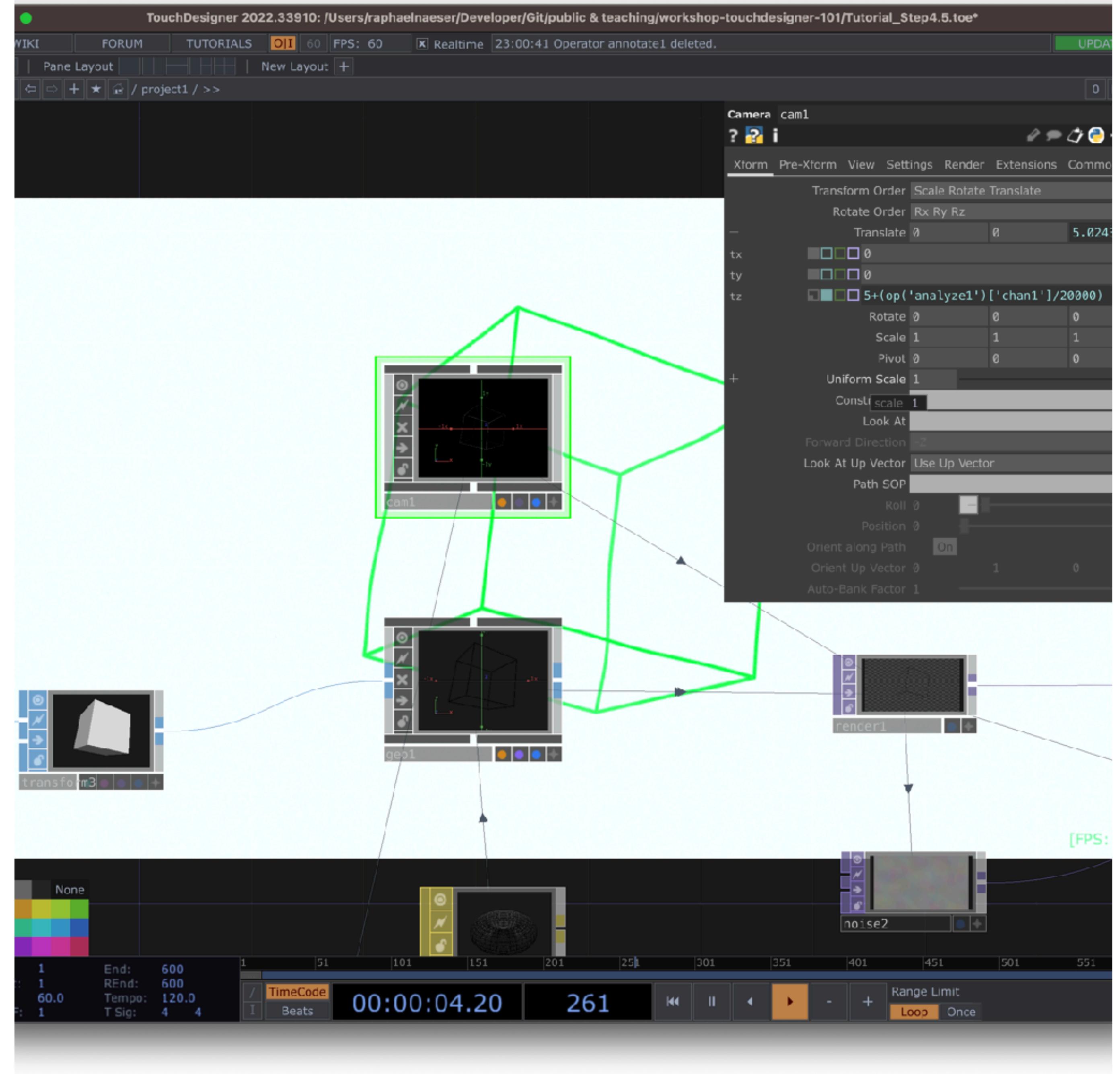
- SOP Transform
- Rz: absTime.seconds\*5



- cam1

- Translate z:

$5 + (\text{op}(\text{'analyze1'})[\text{'chan1'}] / 20000)$



- CHOP Perform

- Rz: absTime.seconds\*5

- DAT CHOPto

- Perform auf Dat ziehen

- TOP Text

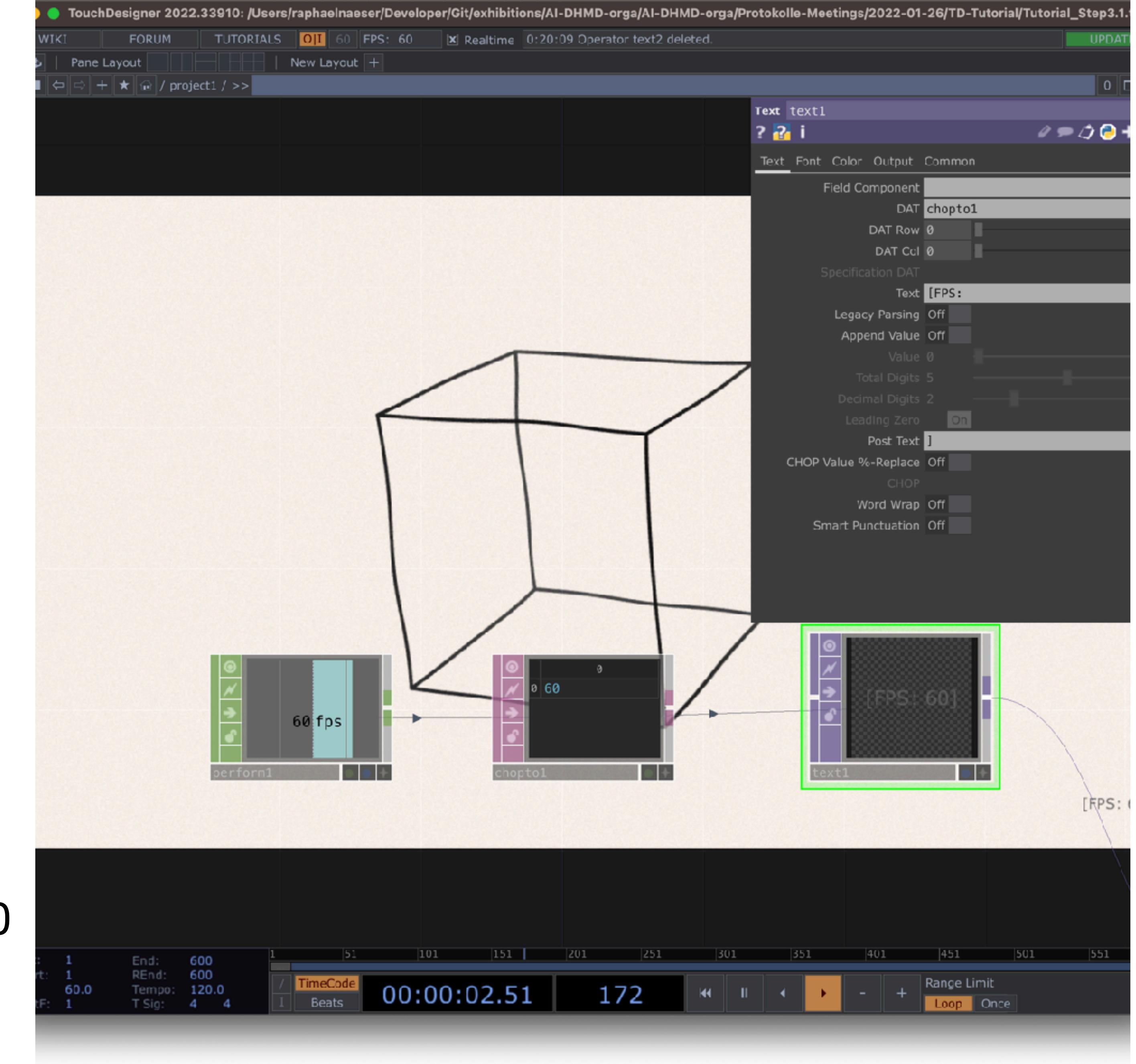
- “chopto1” in Dat eintragen

- Text: “[FPS: “

- Post Text: ]

- Farbe ändern

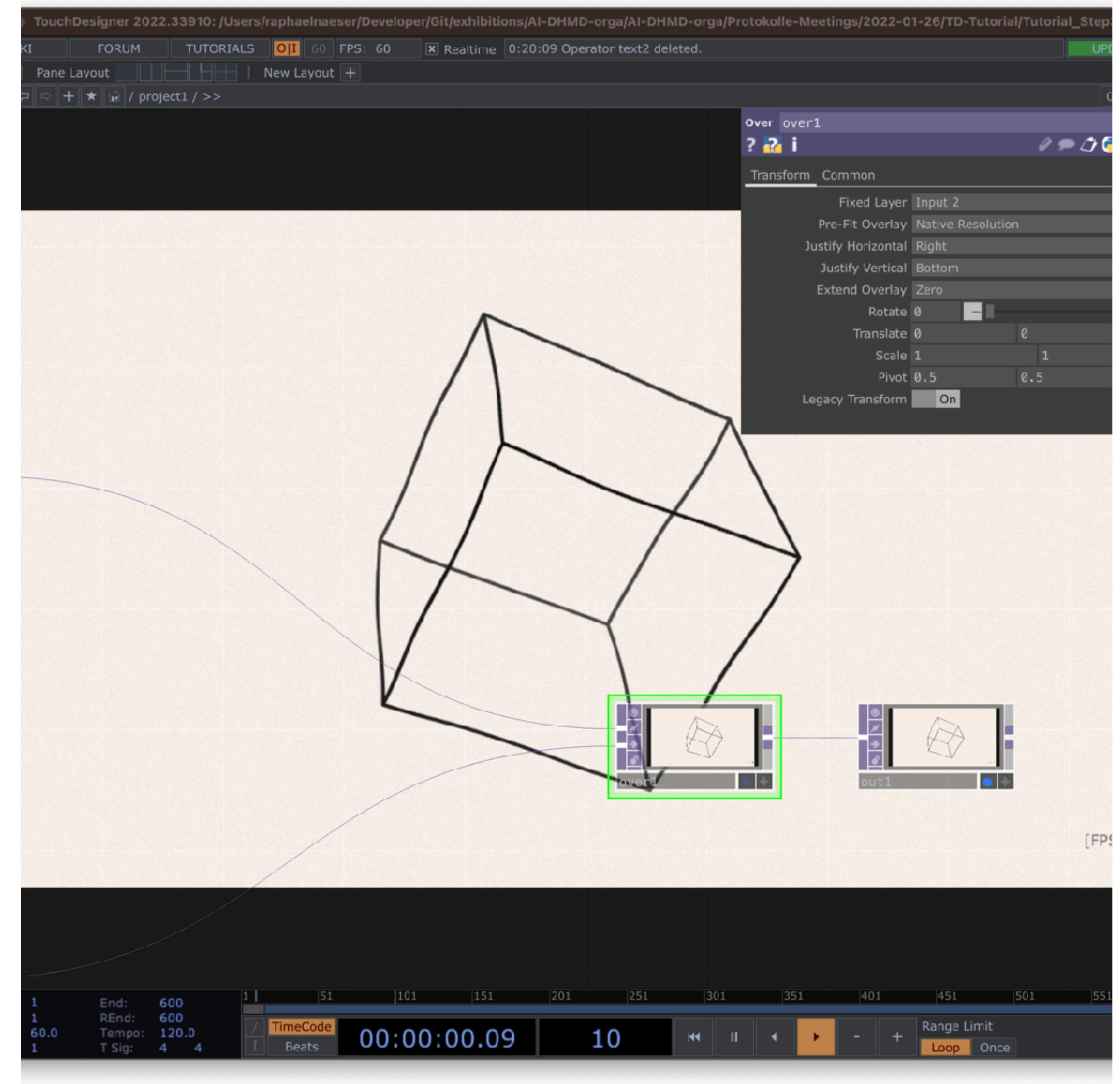
- Commons - Resolution 100x100



## DAT - Daten Operatoren

- TOP Over

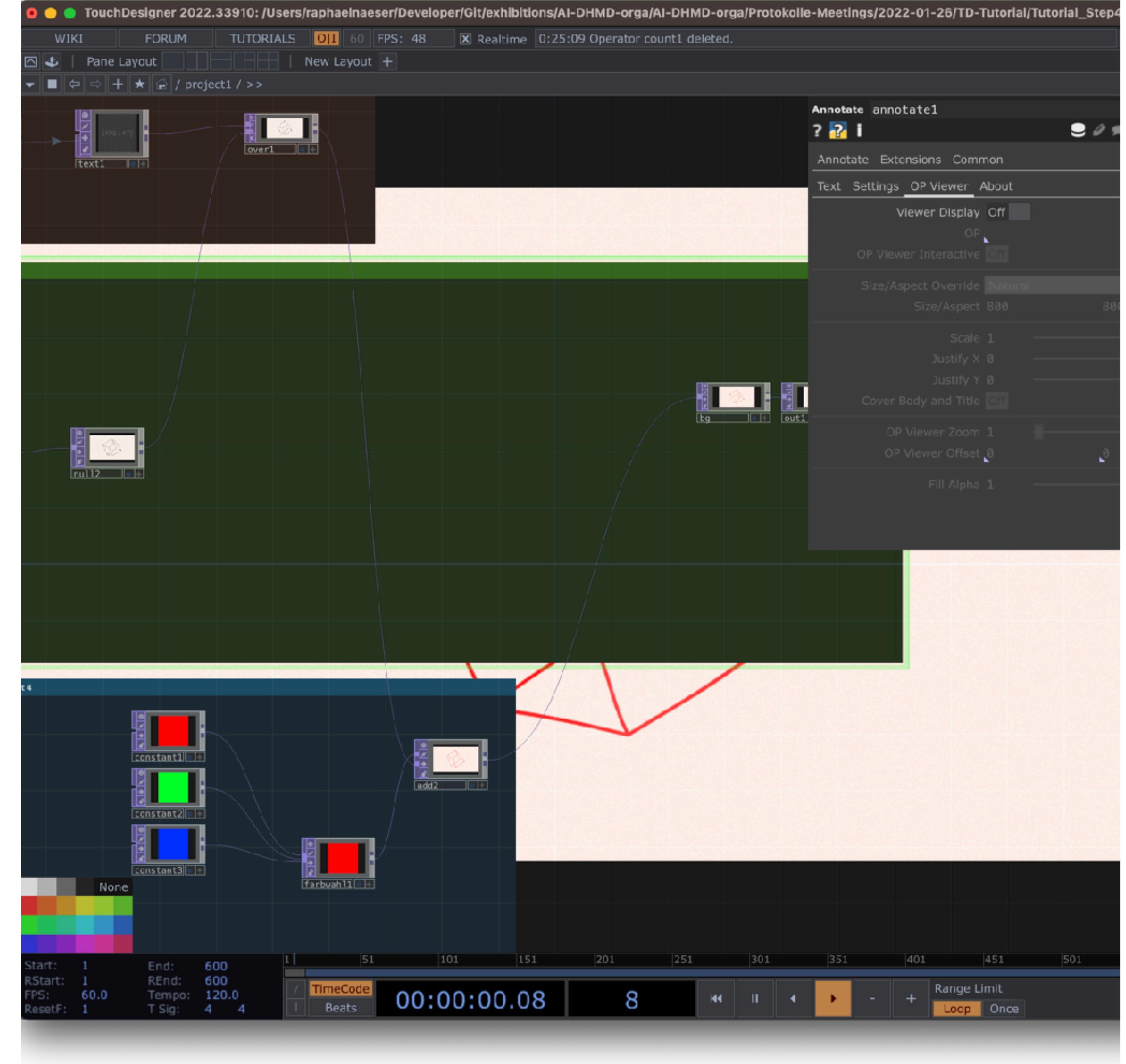
- Unten transform
- Oben text
- Profit Overlay: Nat. Res
- Justify H: Right
- Justify V: Bottom



# **Part 4 - Programmierinterface**

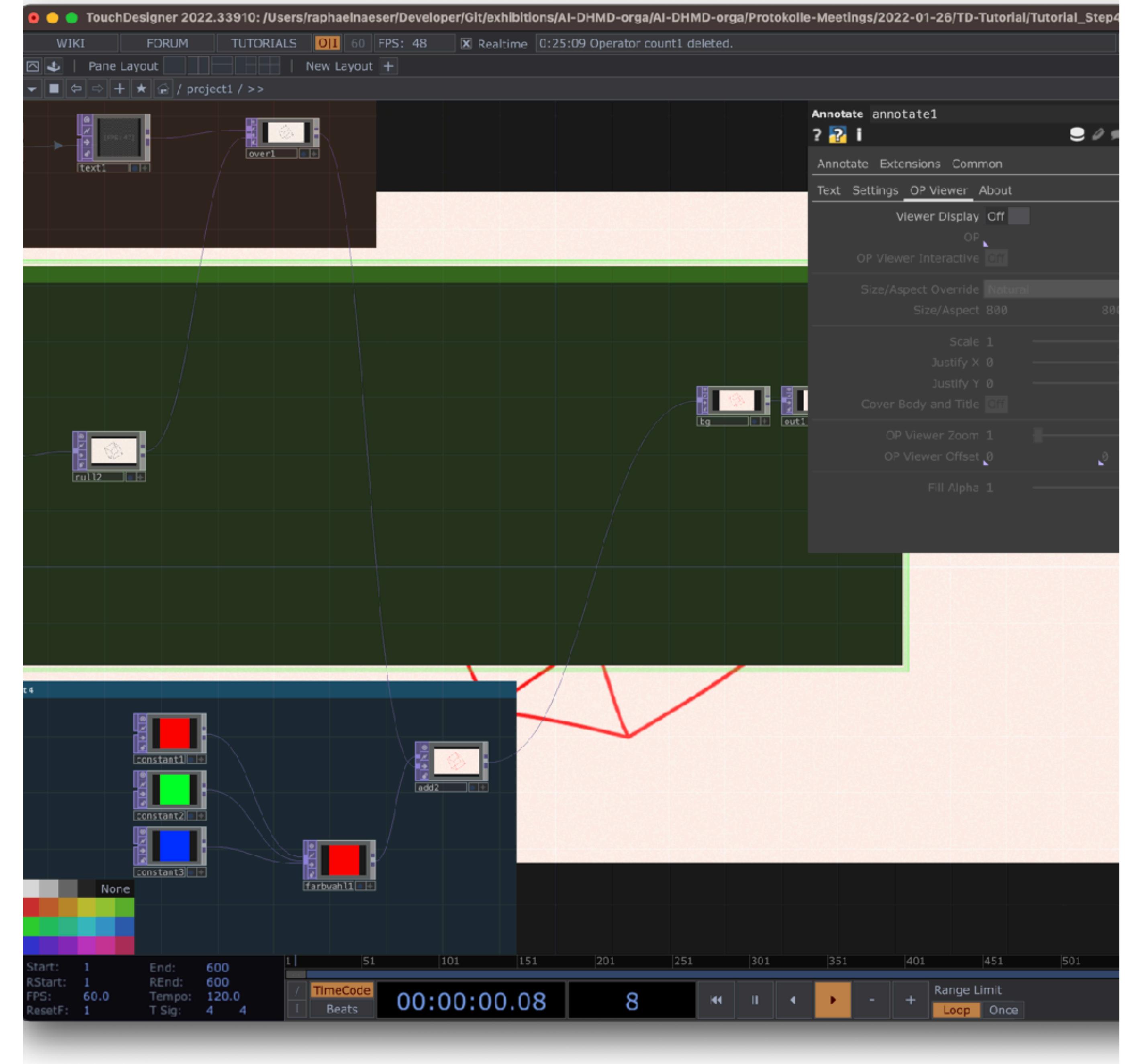
- 3x TOP Constant

- Unterschiedliche Farben
- TOP Switch “farbwahl1”
- TOP add
- Over unten
- farbwahl1 oben



- CHOP Keyboarding
- CHOP Count
  - Limit: Loop Min/Max
  - Limit Maximum:
    - `len(op('farbwahl1').inputs)`
- DAT CHOP Execute
  - Rechtsklick Textport
  - Allen Code bis auf “onValueChange” löschen
  - Code:
 

```
op('farbwahl1').par.index = val
```



- Anmerkungen-, Kommentar-, Network-Boxen mit Rechtsklick
  - Kommentare bewegen Todes nicht
  - Anmerkungen und Network-Boxen bewegen Codes.
  - Können mit dem Node-Farbtoll eingefärbt werden. (Alle anderen Codes ebenso)

