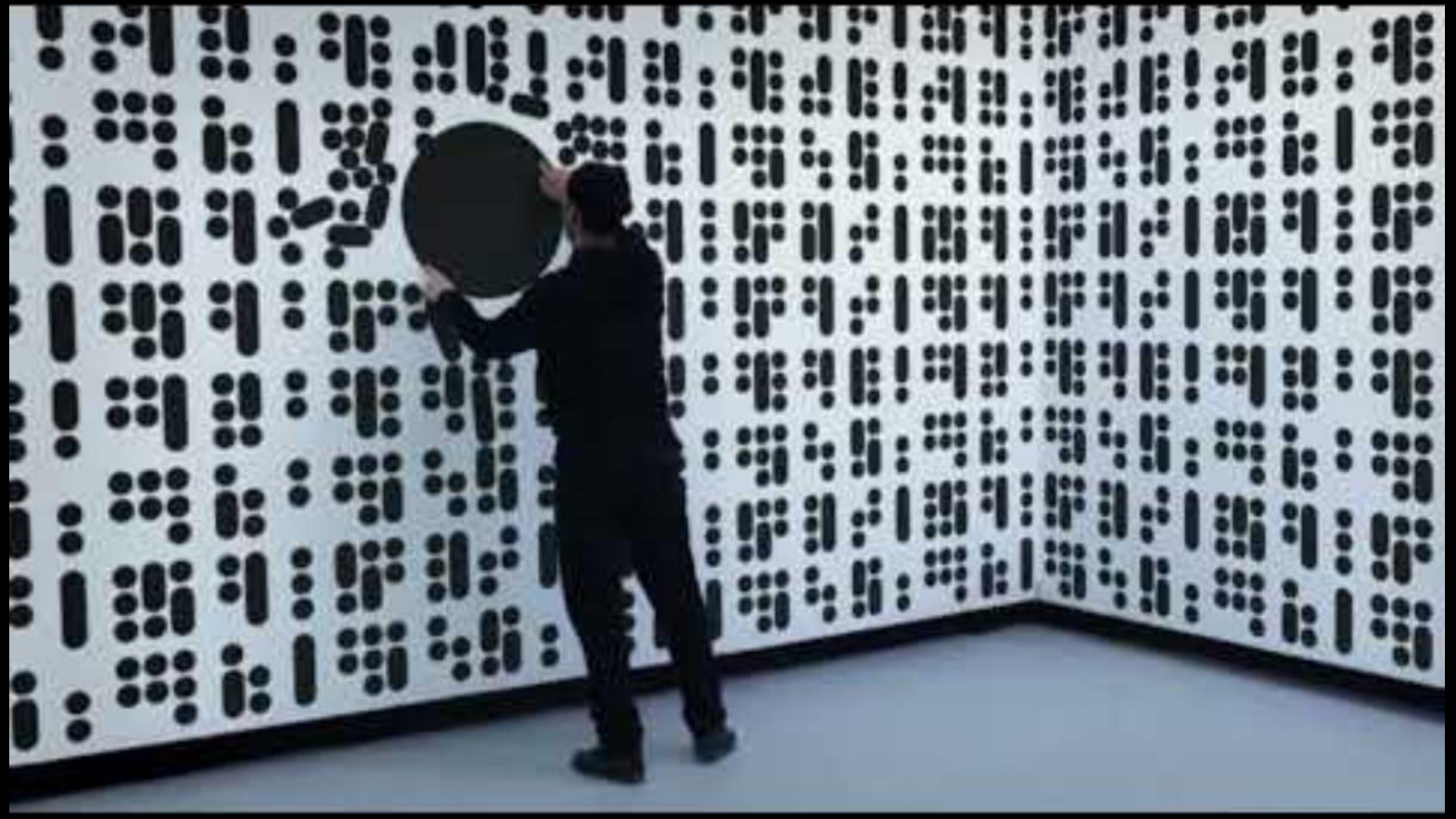
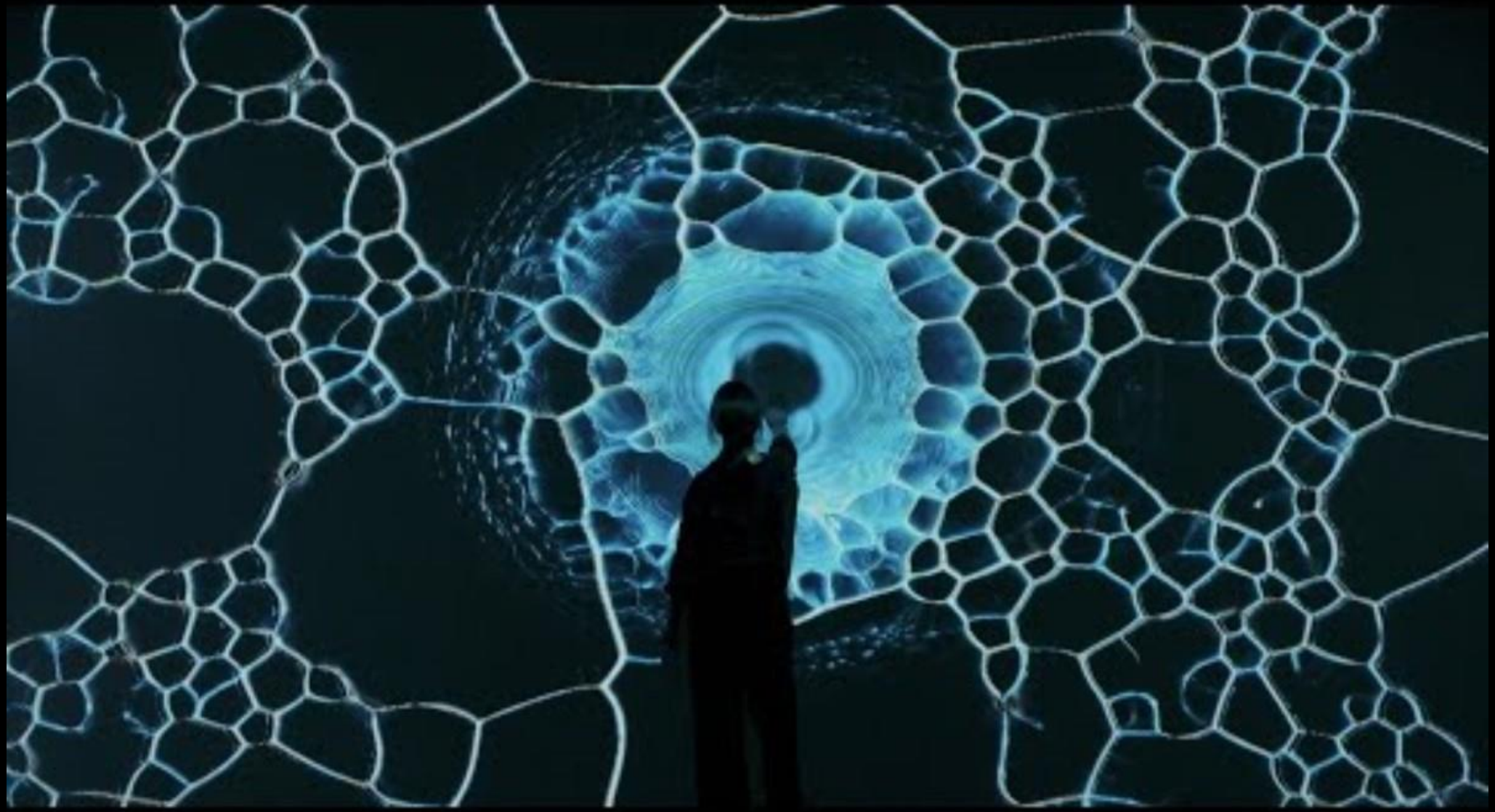
TOUCHDESIGNER BASIC

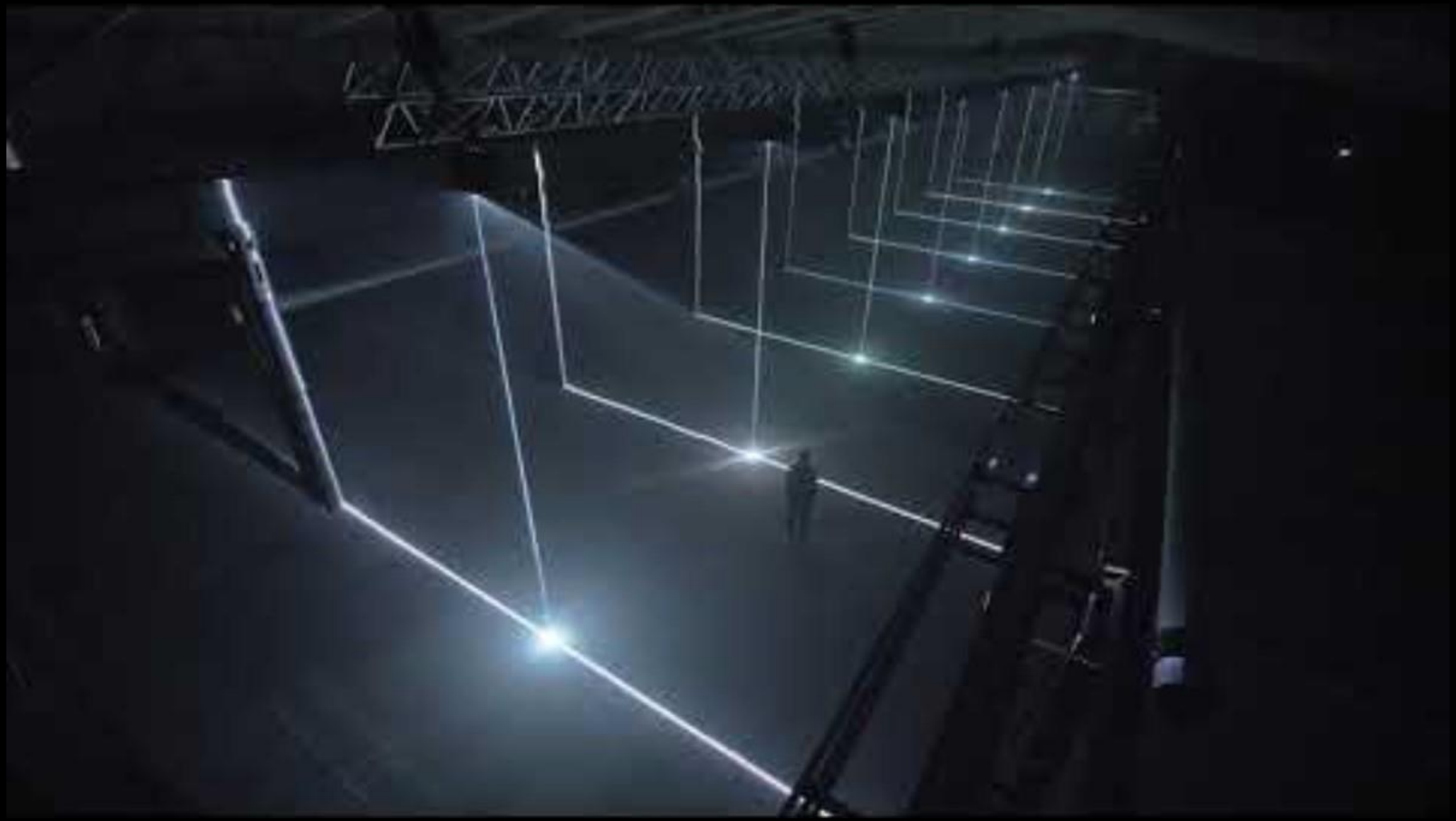
CLASS I

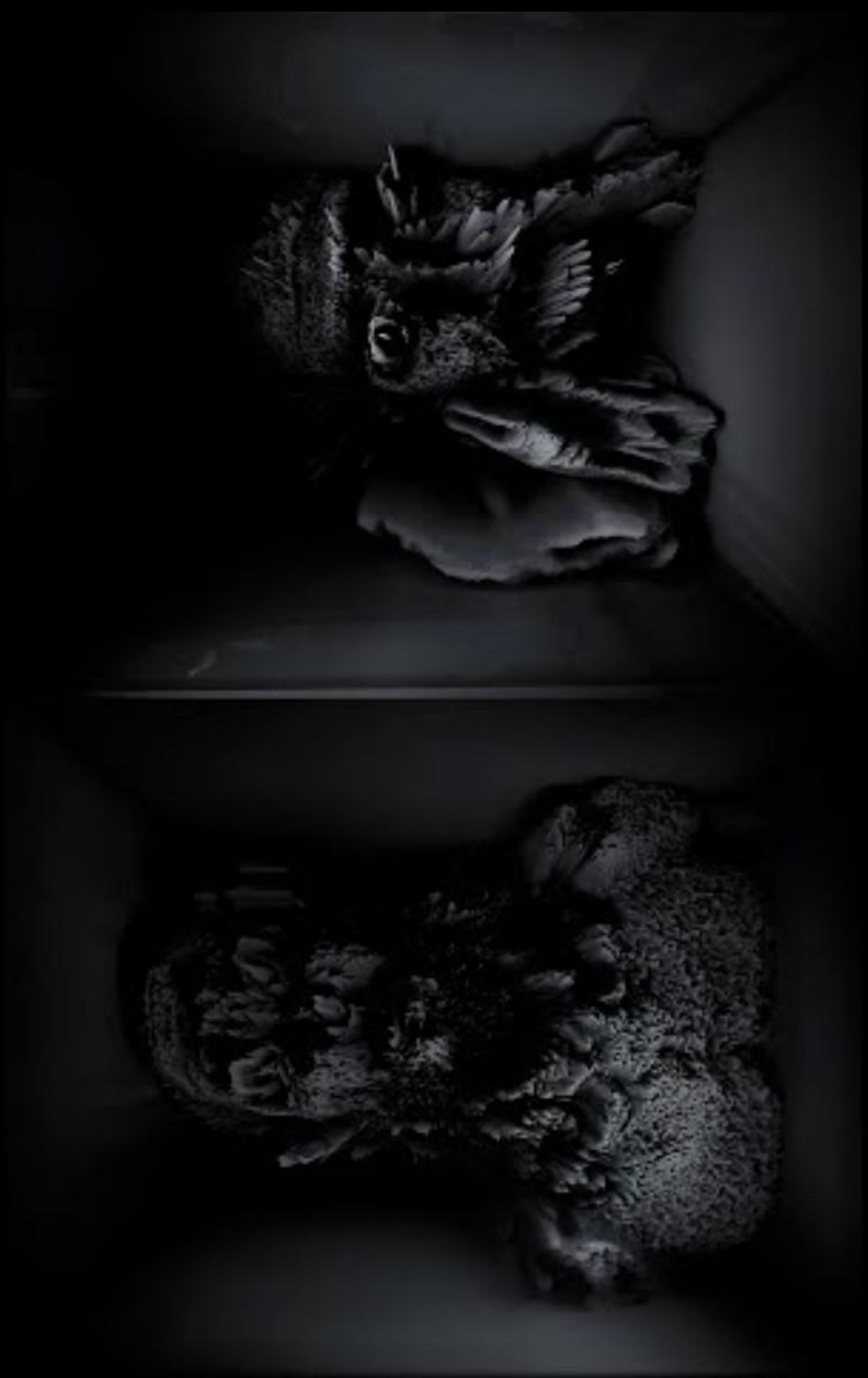
Fasten your seatbelts, enjoy the ride

FIRST START WITH THE CANDIES:





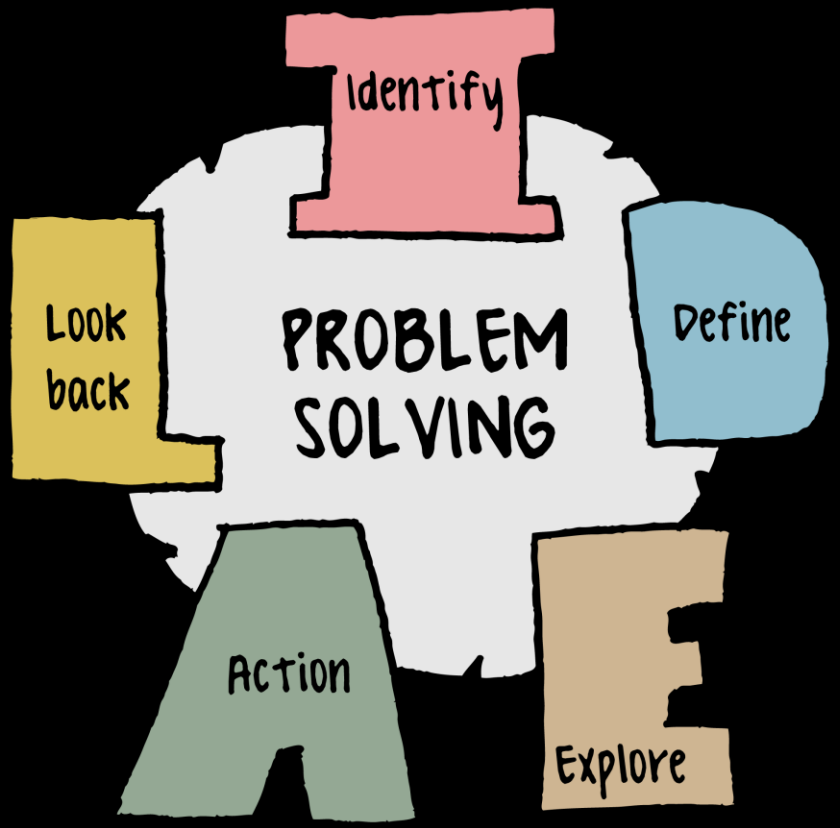




WHAT IS TOUCHDESIGNER?

- Generative video toolkit
- Visual programming environment
- Focused mostly on video
- Interactive installations
- Ableton integration for audio
- Support for several video output formats like, render, streaming, direct output, syphon/spout
- Native support for a lot of different data communication protocols.
- OSC, open sound control, a protocol to send packets of data over a local or extended network
- MIDI, digital sound annotation to control hard and software synthesizers
- Serial, communication over USB between different hardware systems

PROBLEM SOLVING

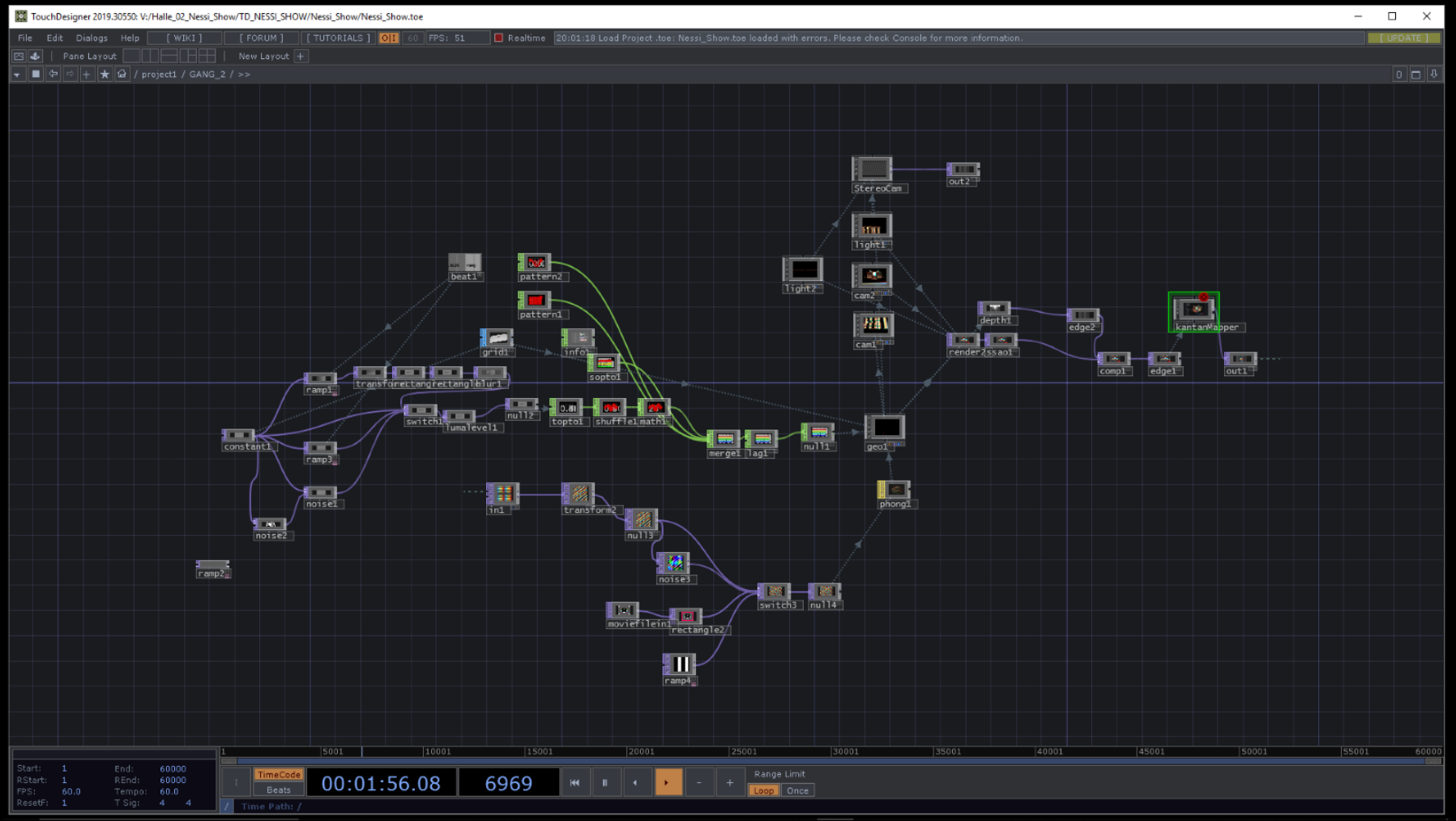


- Touchdesigner is often seen as a video program like Premiere or After Effects.
- But you need a non-linear mindset for working with Touchdesigner and it leans more towards programming or problem solving.
- In class we also work with a problem, or a case, and try to solve it through working with Touchdesigner.

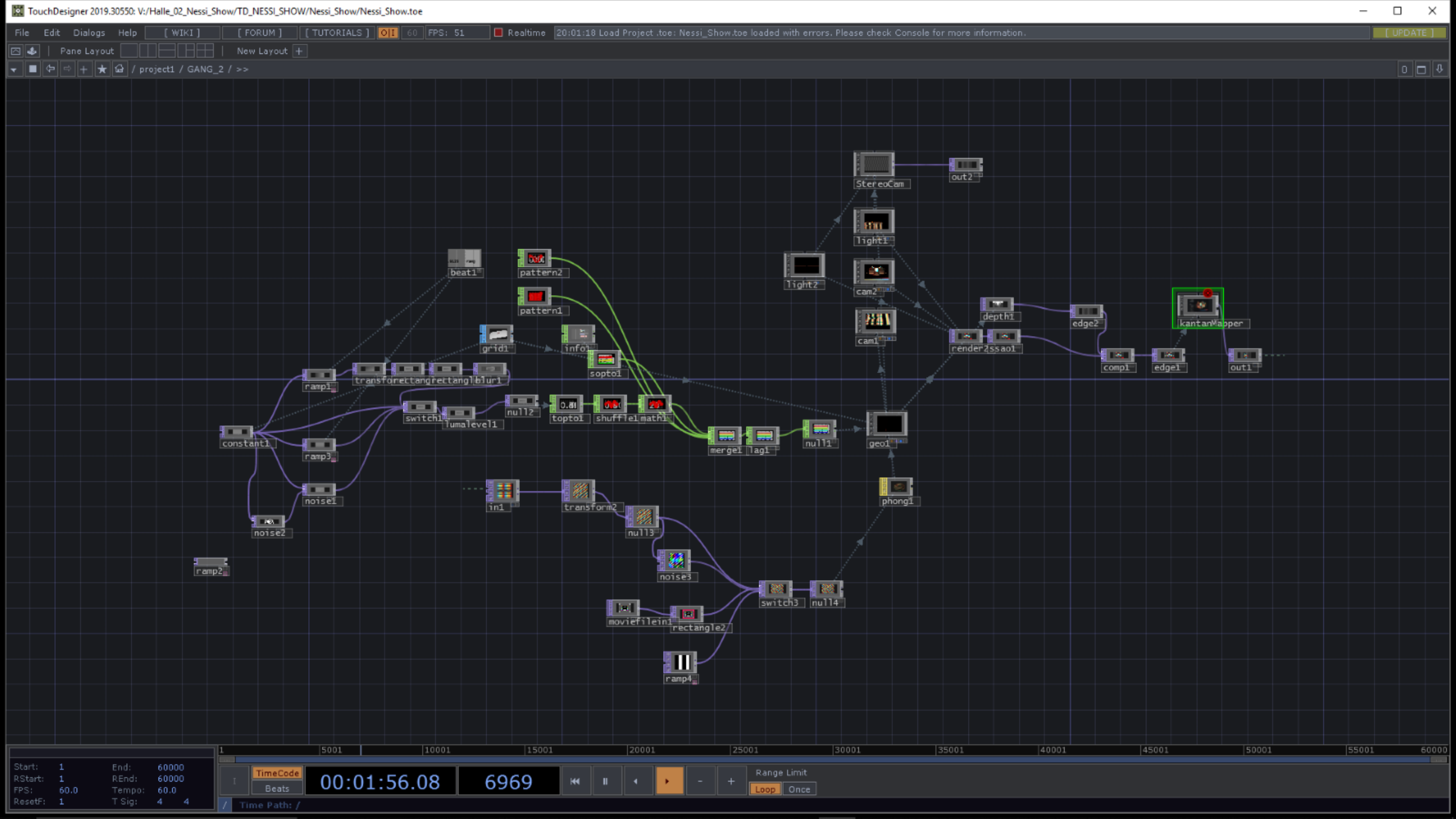
CLASS STRUCTURE

- 1 introduction + basic network
- 2 basic network 3D geometry
- 3 audio reactivity
- 4 MIDI control input
- 5 Data import NASA API
- 6 SPOUT/SYPHON
- 7 open
- 8 open

FIRST LOOK AT THE ACTUAL PROGRAM



11-11-2025

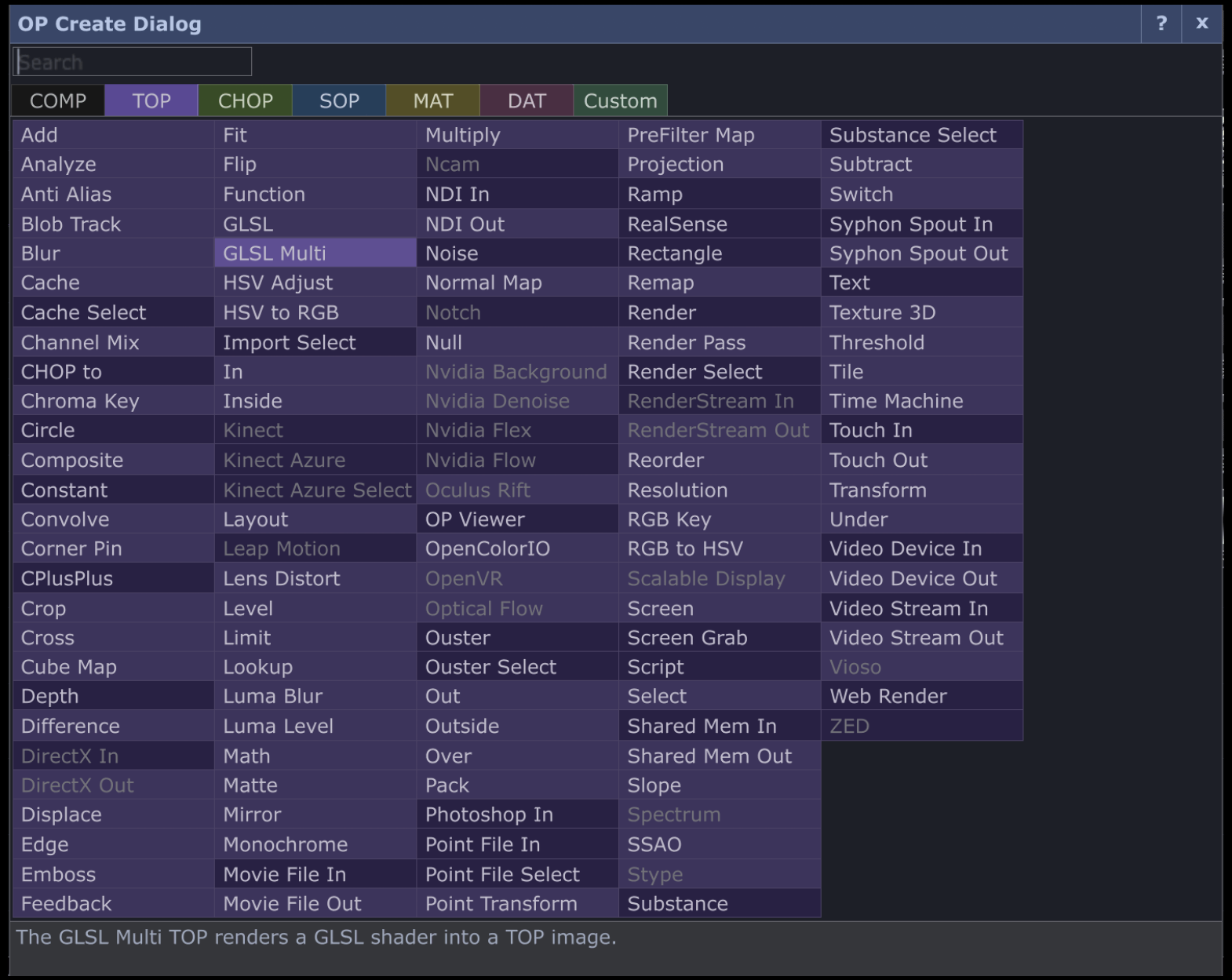

TouchDesigner 2019.30550: V:/Halle_02_Nessi_Show/TD_NESSI_SHOW/Nessi_Show/Nessi_Show.toe
File Edit Dialogs Help [WIKI] [FORUM] [TUTORIALS] FPS: 51 Realtime
20:01:18 Load Project .toe: Nessi_Show.toe loaded with errors. Please check Console for more information.
[UPDATE]
Pane Layout New Layout
/ project1 / GANG_2 / >>
StereoCam
out2
light1
light2
cam2
cam1
depth1
edge2
kantenMapper
render2ssao1
comp1
edge1
out1
beat1
pattern2
pattern1
grid1
info1
sopto1
ramp1
transform1
rectangle1
blur1
switch1
lumalevel1
null2
topto1
shuffle1
math1
merge1
lag1
null1
geo1
constant1
ramp3
noise1
noise2
ramp2
in1
transform2
null3
noise3
phong1
switch3
null4
moviefilein1
rectangle2
ramp4
Start: 1
RStart: 1
FPS: 60.0
ResetF: 1
End: 60000
REnd: 60000
Tempo: 60.0
T Sig: 4 4
1 5001 10001 15001 20001 25001 30001 35001 40001 45001 50001 55001 60000
TimeCode
Beats
00:01:56.08
6969
Range Limit
Loop
Once
Time Path: /

THREE IMPORTANT ELEMENTS

- Operators
- wires
- timeline

OPERATORS

TAB on the keyboard



COMP

- A collection of compositional operators allowing you to add graphical user interface elements such as buttons and sliders. But also containers to store networks in so you can build networks inside networks. Virtual Cameras, virtual lighting and other elements you would need for rendering 3D-geometry into 2D-video also live here. And lastly physical simulators like fluids and gasses can also be found here, note that at the moment those operators work on NVIDIA graphics cards only.

TOP

- A collection of operators that have to do everything with 2D-image and film. Importing a video or image can be done here. Adding video effects can be done here, rendering or sending video streams out can also be done here. People that are familiar with Adobe or other video or image processing programs will probably recognize a lot words like: emboss, corner pin, luma blur, blur, constant. Most of the time these operators will do the same thing as their counterparts in other programs.

CHOP

- If you want to do math, data manipulation, sending or receiving data CHOP is the family you are looking for. Basically CHOP is responsible for high-level data processing based on sample time or computer time. For example, standard one frame passes for 1/60th of a second so the processing of these operators by the CPU are extremely fast. Allowing us to work with audio, creating and editing motion. But also changing parameters in real-time.

SOP

- Operators that work with raw 3D-geometry in the same way other programs do like UNITY, UNREAL Engine, Blender, Cinema 4D and others. You can create scenes and basic shapes like boxes, spheres and cones and you can also manipulate them with operations like cutting, mirror, fillet, twist. You can also import various 3D-geometry files like OBJ, STL, FBX and build scenes accordingly. Note that in order to render or include SOP geometry they have to be combined with virtual lights and camera's found at COMP and a render operator found at TOP.

MAT

- the smallest of all the families. MAT is an abreviation of material and allows you to generate materials in different formats like constants, phongs or wireframes to use in conjunction with SOP geometry.

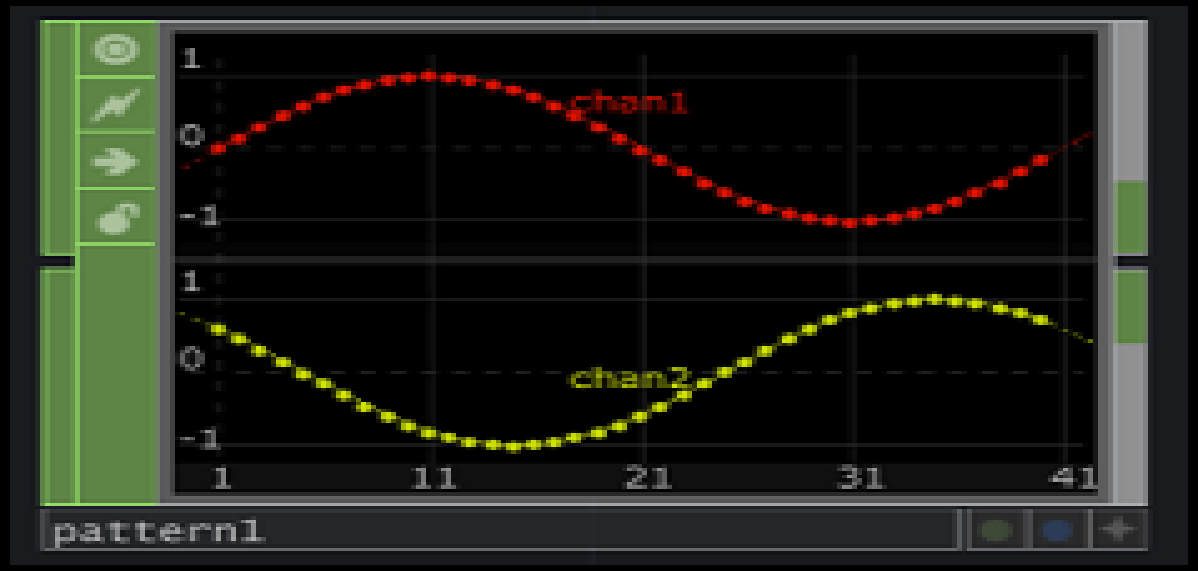
DAT

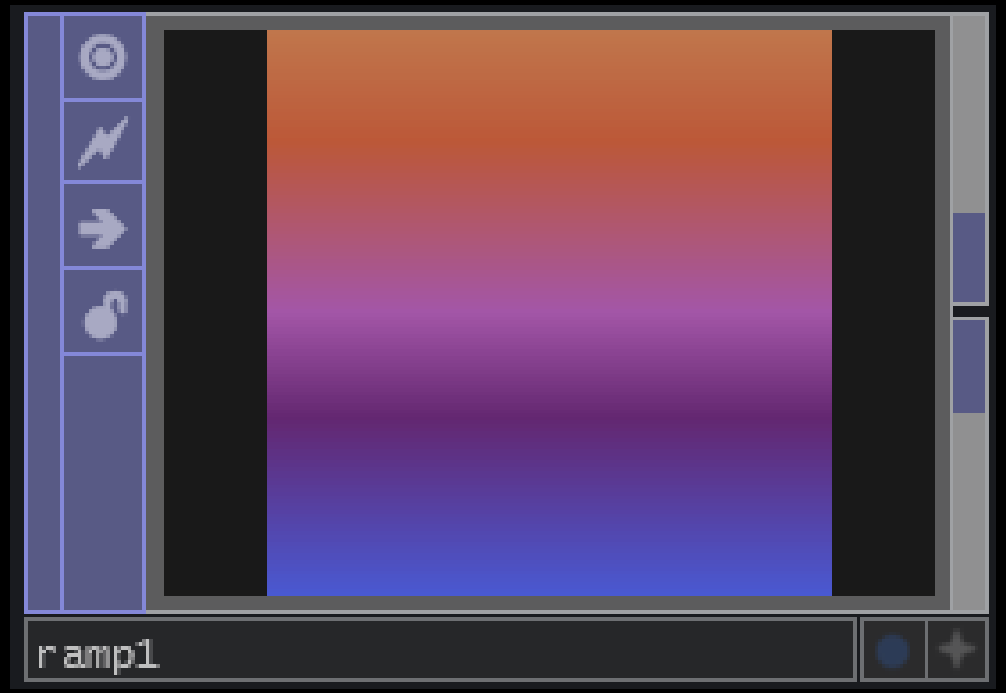
- DAT: are operators to hold text-like data such as strings, scripts and xml's. If you want to execute a script inside TouchDesigner they can be stored, accessed and executed through DAT. Also if you want to access websites, API's or other processes through a static text DATs are your friends. The important difference between CHOP and DAT is that the latter is static and only updates whenever it is told to. Whereas a CHOP will update itself with every passing frame, 1/60th a second, whether or not the value inside stays the same or not. So in terms of importing exernal data into TouchDesigner it is often desired to use DAT because the imported value is being pushed by the external source rather than the clock source inside TouchDesigner itself saving processing power and reducing the amount of unneeded calculations.

SO OPERATORS COME IN DIFFERENT FAMILIES AND ALL HAVE THEIR UNIQUE ATTRIBUTES. BUT DOES THAT MEAN THEY ALL LOOK DIFFERENT? THANKFULLY, NO.

SIMILARITIES

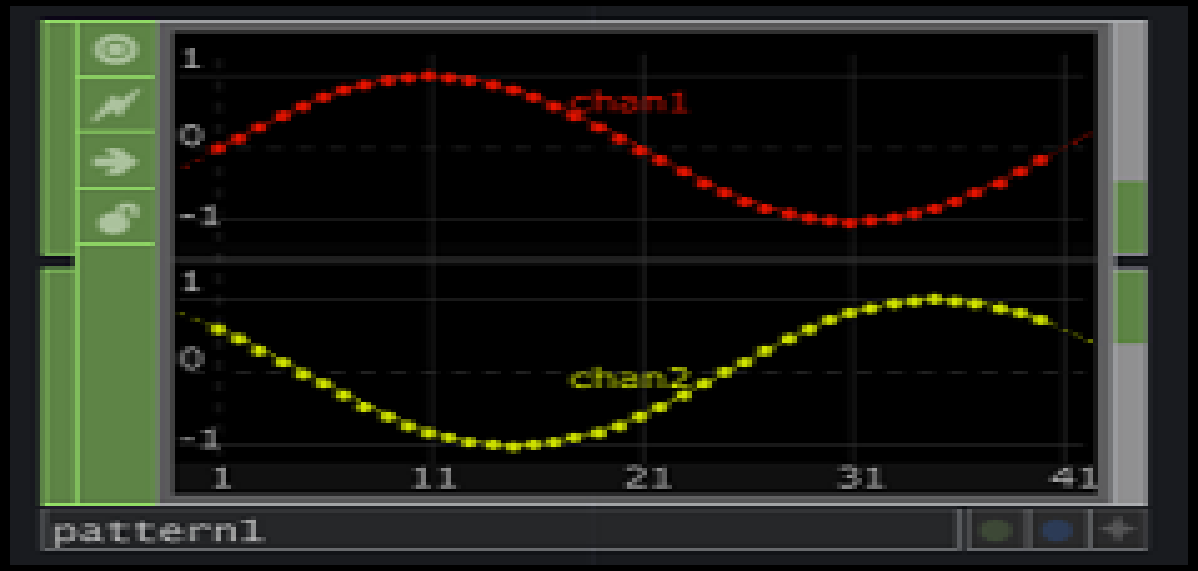
- OUTPUT: Every operator has an output which you can see on the right hand side of the operator. It looks like a black - (minus) sign with the color of the operator above and below it. When you click with the left mouse button on the output a wire will be connected to the output and the cursor allowing you to connect to an input of a different operator in the same family.

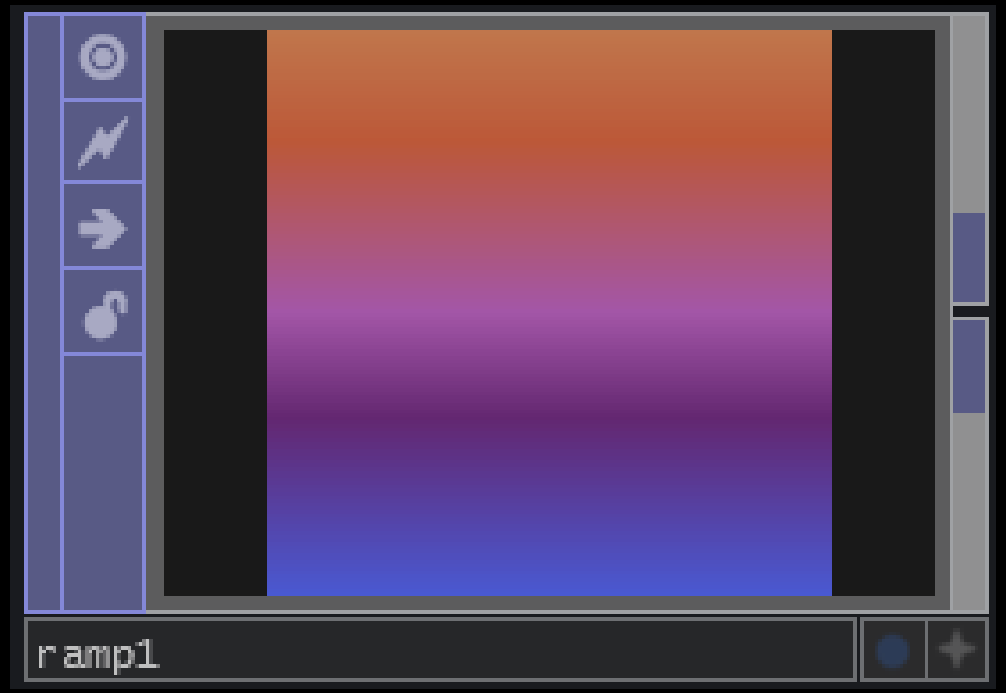




DIFFERENCE

- INPUT: the input is situated on the left hand side of the operator and can be seen on the CHOP (green) operator above but is missing on the TOP (purple) operator. The reason why the input is left and the output is right is because information in TouchDesigner flows from left to right to strengthen readability of the network. It is also adviced to place operator left to right and not top to bottom, for hygiene sake. What does it mean if an operator is missing an input? Simply that it is the start of a network or a branch inside your network.

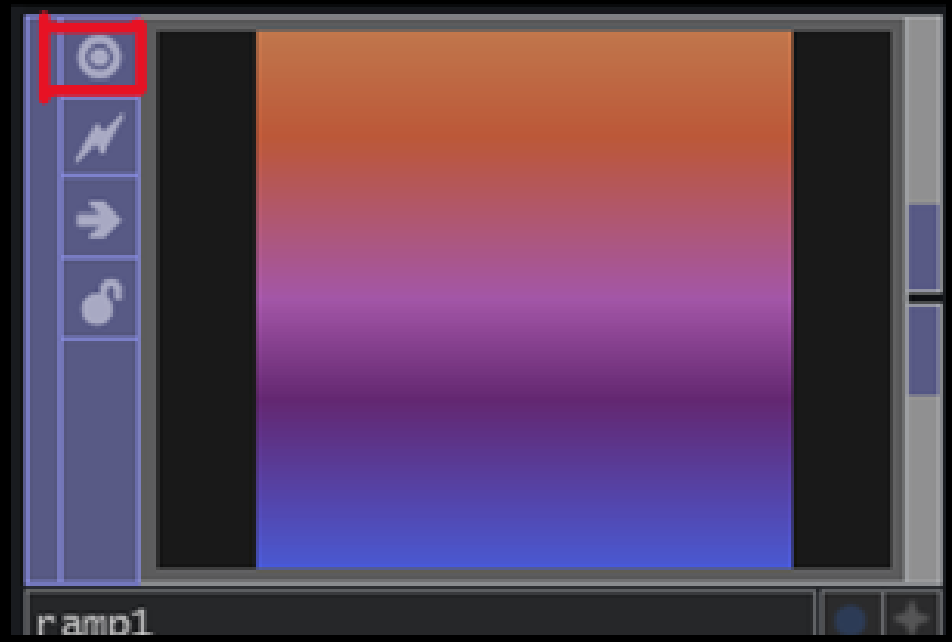




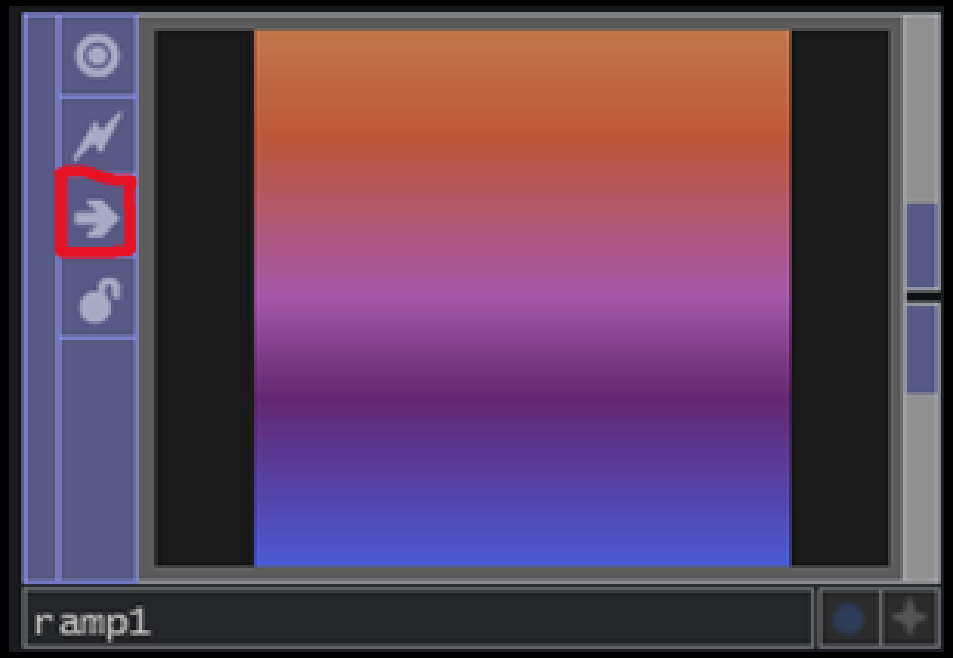
THEN ALL OPERATORS HAVE ‘FLAGS’ WHICH ALLOW YOU TO ‘FLAG’ AN OPERATOR SO IT DOES SOMETHING ADDITIONAL INSIDE THE NETWORK SPECIFIC TO THE OPERATOR ON A GLOBAL SCALE. MEANING EVERY OPERATOR CAN DO THAT INSTEAD OF THE OPERATOR SPECIFIC TASKS. THERE ARE A NUMBER OF FLAGS BUT THREE OF THEM ARE MOST IMPORTANT IN OUR CONTEXT AND WILL BE USED THE MOST.

VIEWER FLAG

- Viewer flag: this button will disable the viewer of the operator. Meaning the little preview allowing you to see what each operator does, in this case of the image the blue to orange gradient. When building your network generating the viewer for each of your operator in real-time can become very CPU consuming very fast. To free up some space and give your cpu a break disable the viewer and only enable it to check on your operator.



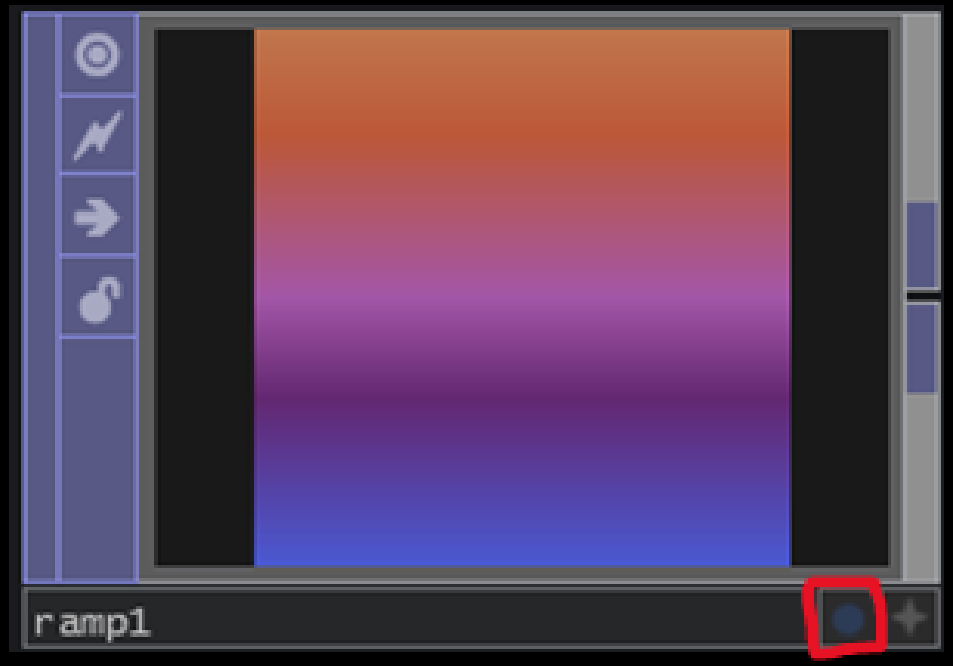
BYPASS FLAG



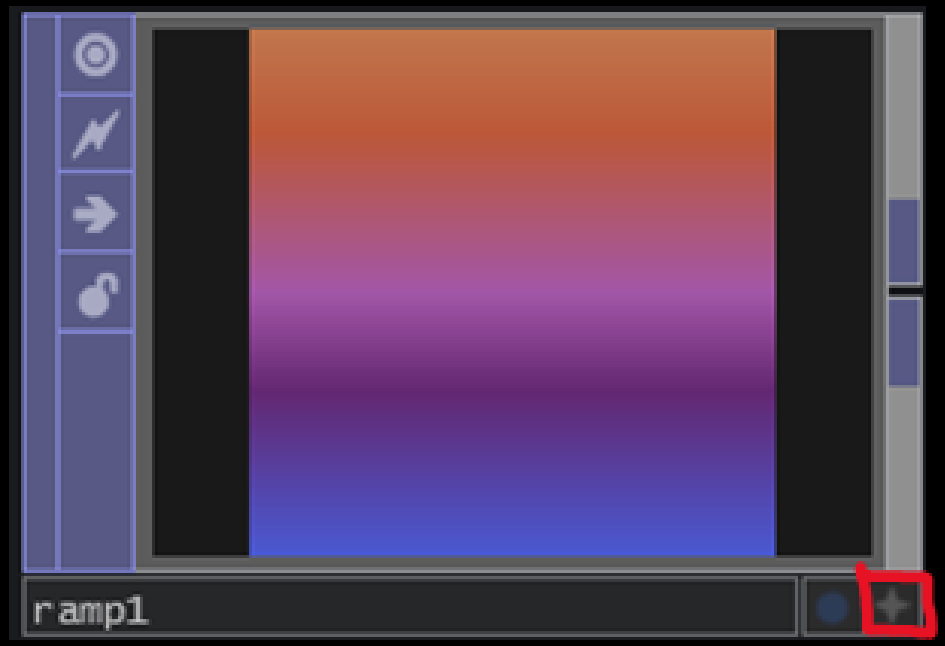
- Bypass flag: The bypass button will set the operator in a pass through state only passing through the received information and not adding its own operation. This is handy to momentary pause certain operations for bug hunting but also to curate whether or not you like a specific effect.

DISPLAY FLAG

- Display flag: this small blue dot will display the content of the viewer of the specific operator onto the background. Very handy to do with the last operator in your network, the one that receives the end-result, to have it display on the background so you can see a bit better and larger what is actually going on inside the network. Note that if you make an operator display on background that is situated in the middld or beginning of the background you will not see the most updated version of your network. You can however have multiple operators display to the background.

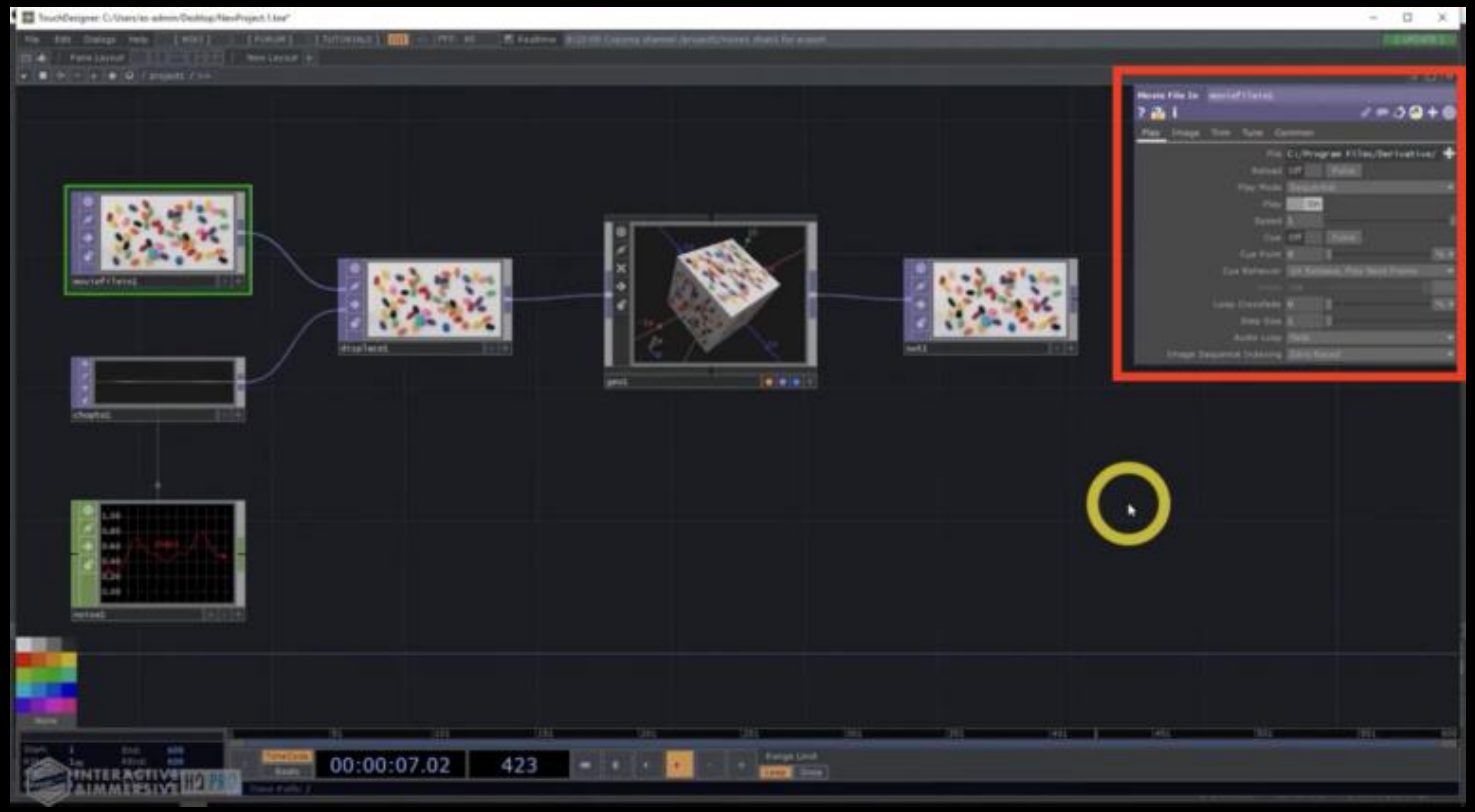


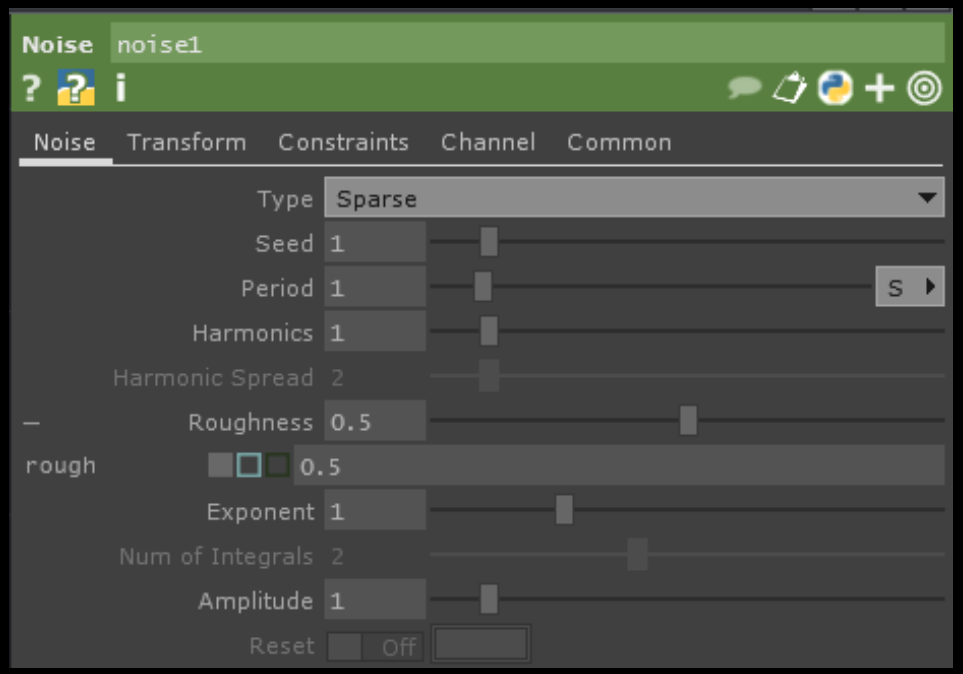
VIEWER ACTIVE FLAG



- Viewer active flag: when enabled it allows you to interact with the viewer of the operator. In 3D-geometry it allows you to pan around and interact with the geometry without actually transforming the geometry. With CHOP, TOP and DAT enabling the viewer active flag allows you to click and drag the content onto another operator, also outside families, to reference one another. We will use this a lot, even in first class and it is advised to get familiar with this as soon as possible.

PLACING AN OPERATOR WILL ALSO OPEN THE PARAMETER WINDOW





- The parameter window is a collection of sliders, number boxes and settings for you to play around with and change the specifics of a certain operator. Every operator has its own unique parameters and most parameter windows also have multiple tabs. Make sure to play around with them, test them out and see what happens. In here is where you truly make the network your own. Although every parameter window is unique almost every one of them has a ‘common’ tab in which you can change things like resolution and pixel format. If you want to free up some space in your network for visibility you can press ‘P’ on your keyboard to hide and show the parameter window.

EXPERIMENTATION AND FAILURE IS ADVISED AND MANDATORY!

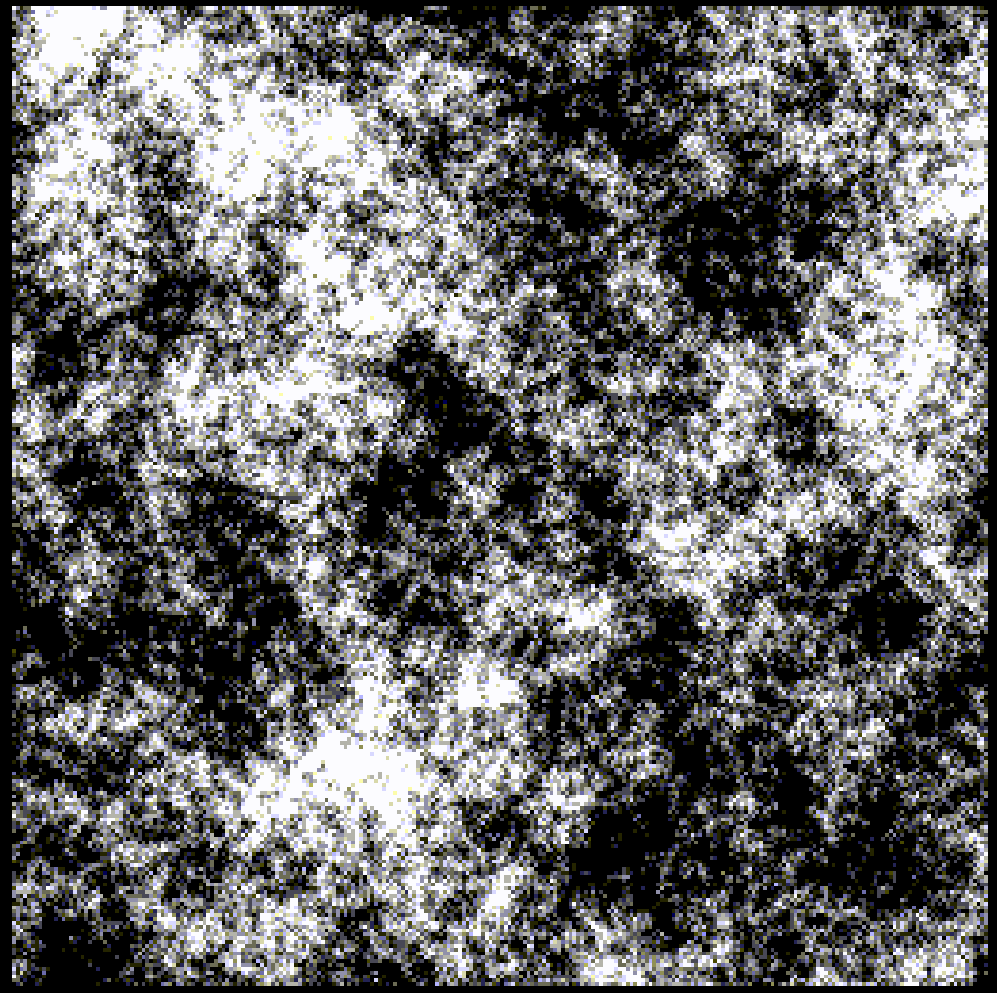
- When opening up the OP Create Dialog you can see immediately what operators there are in which family. The name of the operator will mostly tell you already a lot about the function. When hovering over the different operators a brief explanation of the operator is given in the OP Create Dialog. This helps tremendously with finding the right solution to the problem you encounter and try to solve. Just open the window, read the discriptions and explenations and just try it out, fiddle around with the parameters. In worst case it doesn't do what you want it to and you just erase it again and repeat the steps with a different option this time. Really that's how all the pro's do it also.

CASE: THE ROTATING BANANA

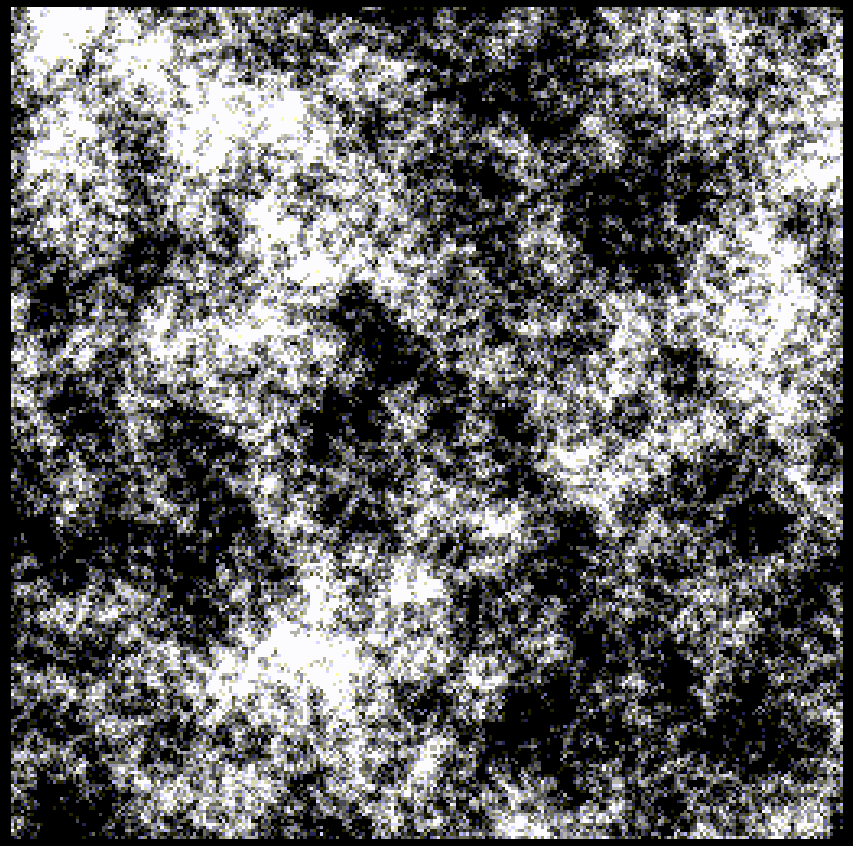
BREAK UP THE DIFFERENT ASPECTS

- We need a banana
- We need to make it rotate: 1. we need some control over the image. 2. we need a way to automate the control.
- We need to export from Touchdesigner into a short GIF

CASE II : ANIMATED NOISE



BREAK UP THE DIFFERENT ASPECTS



- We need an image noise generator
- We need to make it evolve: 1. we need some control over the image. 2. we need a way to automate the control.
- Tip: evolving the noise goes by transforming it over the Z-axis
- We need to export from Touchdesigner into a short GIF

SMALL PYTHON EXPRESSIONS

- There are a lot of ways to animate things in Touchdesigner. One of the ways is to use small python expressions like this one:
- absTime.seconds*1

absTime.seconds*1

The expression can be explained as follows:

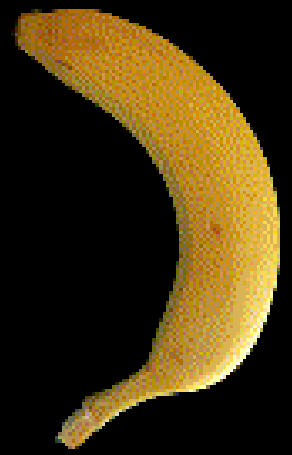
absTime stands for absolute time, which means the flow of time measured by the cpu in your computers which acts like a master clock for all the different programs and applications you use.

From the calculated absolute time we want to use the measurement of seconds. Hence the absTime.seconds.

Finally the last part, *1, means for every seconds passed on the absolute time we are going to add 1. The number after the asterisk, *, is a numeric value only. We could also say absTime.seconds*100 to add 100 for every second passed or absTime.seconds*0.01 to add 0.01 for every second passed. Resulting in an increase and decrease in speed respectively.

CASE III : SWITCH IT UP WITH KEYBOARD CONTROL

BREAK UP THE DIFFERENT ASPECTS



- We need our banana and noise networks
- We need a switching operator that can switch between the networks
- We want keyboard input
- We need some kind of counter or limiter so we can use one button for switching between two or more networks.