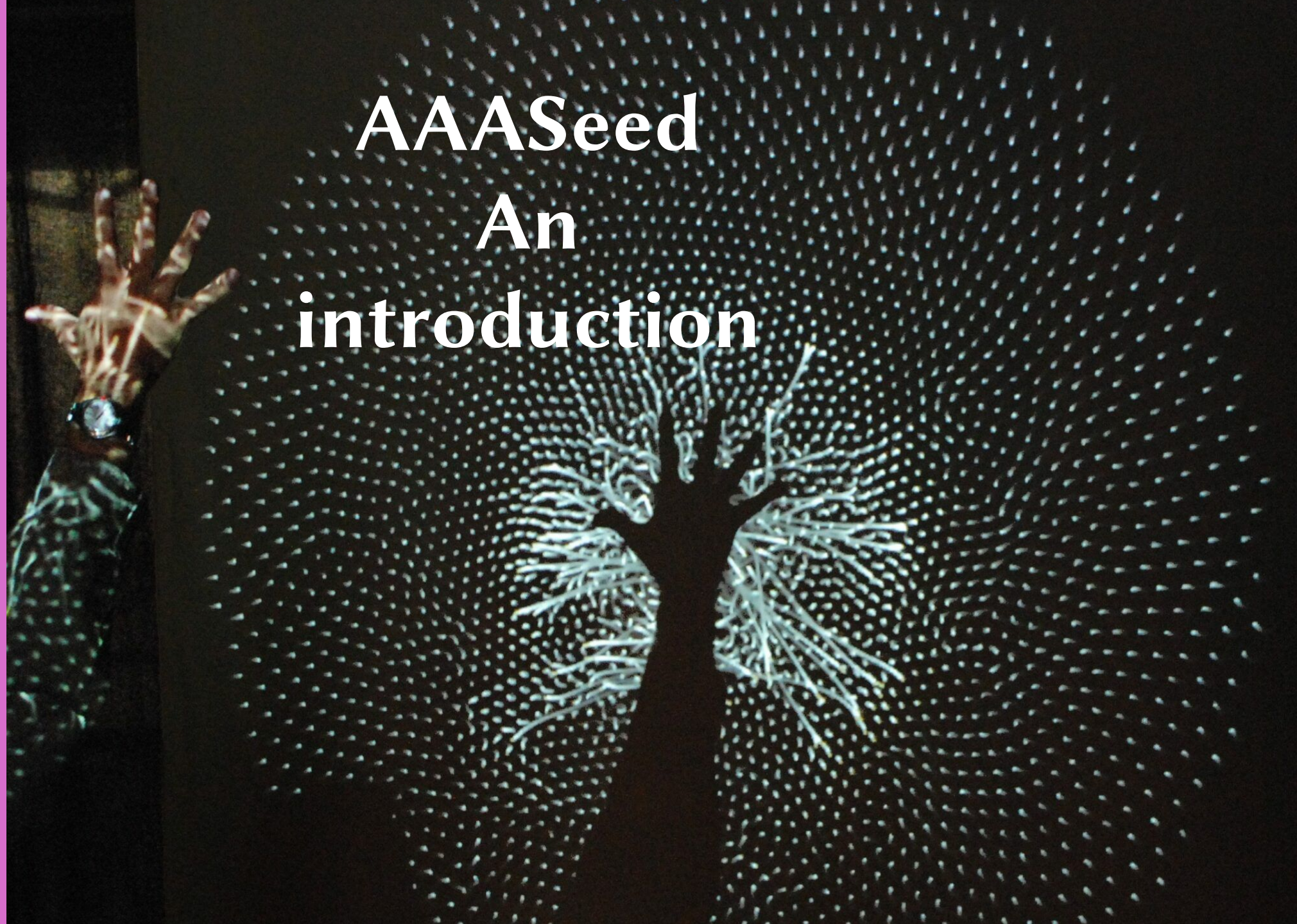
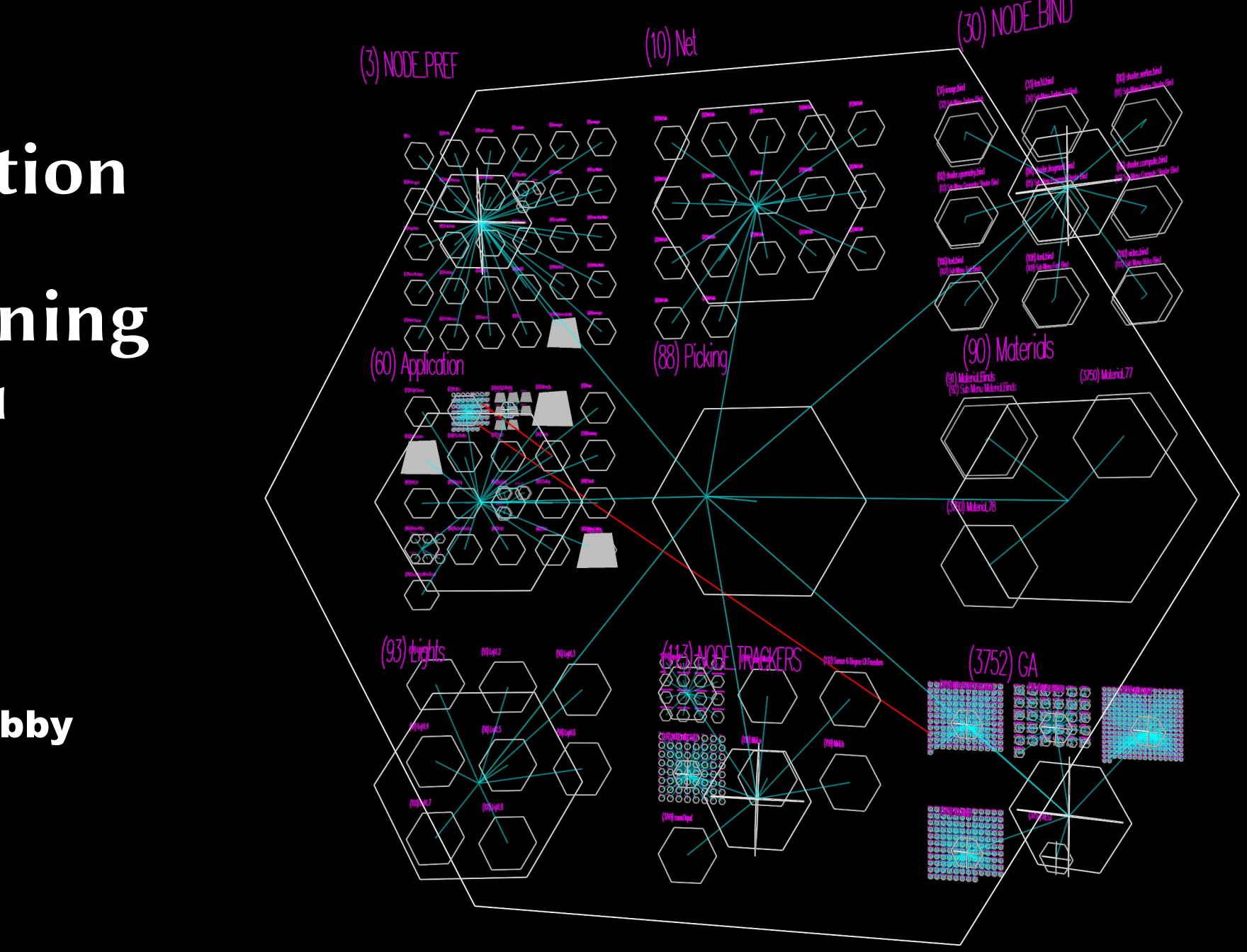


AAASeed
An introduction

AAASeed
An introduction
Part 1:
In the beginning

- What is AAASeed
- Point of View
  - **Artist**
  - **CCI**
  - **Developer**
  - **Education, Hobby**
  - **Usert**
- Ga Bu Zo Meu



What is AAASeed ?

- A **generic toolkit** for building **real-time** processes
- Designed for:
  - **V**ideo **J**ockeying & Live Mapping
  - **Open data** visualization
  - **Interactive Art** installation
- **Open-source** soon (MIT license), **free**
- Created by **Mâa,** with great help from **Franz Hildgen**
  Refined over **25 years real-world use**
- Stabilized and improved through a European Project

  https://ArtCast4d.eu

  Early Adopters Program
- **AAASeed is build to last**

  https://AAASeed.org

For whom is AAASeed ?

Artist

- **Compose** and **assemble** blocks on the **fly.**
- Most blocks handle **graphics processes.**
- Others can manage anything, for example:
  - **input** (cameras, sensors)
  - **Image analysis**
  - **Connections between elements**
- Create **live, graphic** and **interactive outputs.**
- **Free License**

  free digital support for generative/interactive artworks.

Who is AAASeed for?
Creative Cultural Industries

- Video professionals
  Control and display videos, images, and text.
- Immersive environments
  Multi-screen and multi-machine setups.
- Works live with existing tools:
  - Protocols: Spout, OSC, MIDI
  - Software: TouchDesigner, vvvv, Unity, Unreal, Chataigne, OBS studio...
  - Sound software: Max/MSP, Ableton Live, PureData/PlugData...
- Use cases:
  - Dance, Theater
  - TV, Digital signage, Interactive screens
  - You name it
- Free License
  Keep your budget focused on production.
  Long time warranty on exploitation.

For whom is AAASeed ?
Developer

- On the base of an old school **robust C++ rendering graph**
  using the **very fast and flexible Lua scripting language**
  you can access the low level functionnality:
  Window system, custom C object, OpenGl,
  C++ objects, Gl Shaders, OpenCl, OpenCV,
  Nvidia Flex, dlib, bullet, ...
- A rich **lua virtual machine provide**
  an **interface** to edit and customize realtime processes
  most of it happen **live** while processes are running
  extend and edit on the fly the interface

For whom is AAASeed ?
Student, Hobbyist, Teacher

- Have a **rich, fast and complete tool**
  to **script graphic ideas on the fly**
  learn and understand 3d graphics
  experiment and test before going further
- **Use shaders**
- **Live coding**
- **Teach coding and graphic**
- **Experiment Mathematics**
- An **equivalent to processing** ?
  Need probably some graphic primitives to be added
- Infrastructure for research ?

User point of view

- This a W**indows** application for now

  AAASeed_Metal.exe

- Tested against **windows 10 and 11**

  but should also work on Windows 8,7 even XP.

- It can **run on low end machines**

  even an executable for non Avx2 Processor (AAASeed_Wood.exe)

- But it likes fast machines and **uses the Gpu a lot**

  Graphic Processor Unit
  Loves **NVidia** but functions on integrated Intel
  supports **Amd** most of the time (getting better on Amd every month)

Shadoks point of view

"In case of problem just pump"

- talking with the **GA**

  Global **A**ction: the top level

- using **BU**

  Box **U**ser: element of interface

  BUS contain one or several BUs

  a BU can contain a **BUS** and so

  this way a BU can contain other BUs

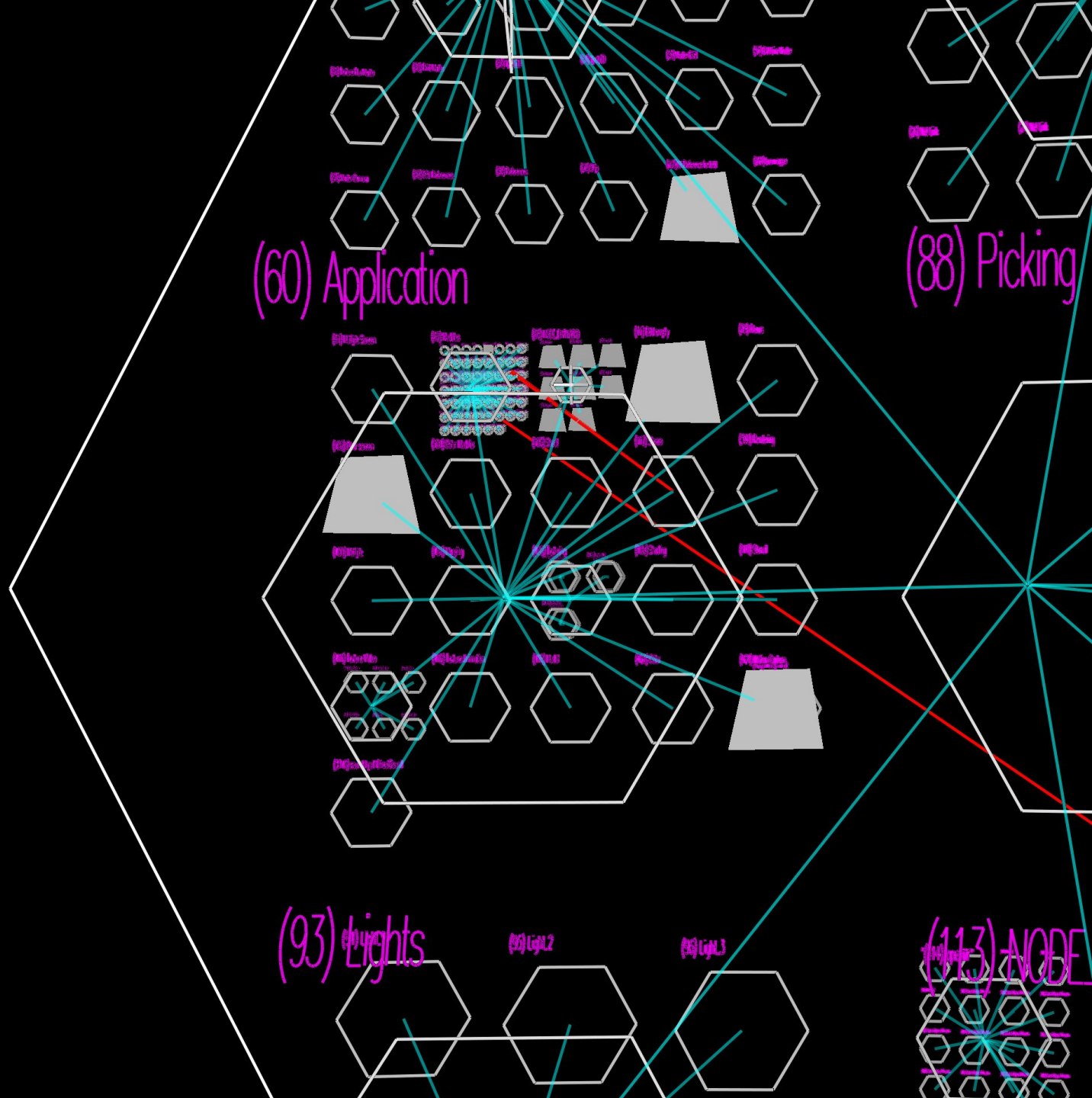
- **Using MEU**

  Module **E**ditable **U**nit: functional editable blocks
  sometime represented in their short form the **MU: M**odule **U**nit

- We lost the **ZO** but we will find it, promised («On a perdu les ZOs»)
- More on this: search Shadoks on YouTube
  - https://www.youtube.com/watch?v=Sla57Zw-FN4
- Thanks to aaaproduction (https://www.aaaproduction.fr/)

AAASeed
An introduction
Part 2:
Install

- Setup AAASeed on your computer
- Folder structure
- AAASeed Sources



Install 1/4: Required software

- **Klite Codecs**

  handle decompression of video (AAASeed use it through DirectShow until 2025 September at least)

  https://www.codecguide.com/download_kl.htm

  we use klite Mega, but other options should function too

  installed with default options all the way

- **7z**

  handle compression and decompression of 7z files

  https://www.7-zip.org/

  Windows 64-bit x64 version

  7z ... -x64.exe : execute to install

- **Visual Studio Code**

  text / code editor used by AAASeed by default

  NotePad++ and Sublime Text 2/3 were used and should still function
  Flatland/Pref/MASTER/Lua/lua_master/editor

  https://code.visualstudio.com/

  Download for windows

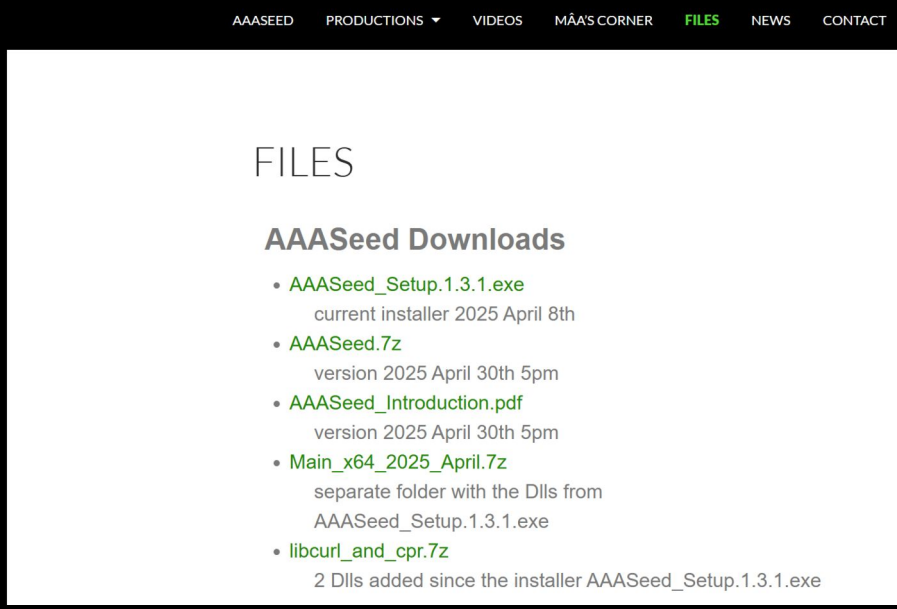
  Use extensions to color code for lua, GLSL, OpenCL

  Use a workspace with the complete AAASeed folder is very practical to search and find.

Install 2/4: Other software

- **PureData and/or PlugData**
  https://puredata.info/ and/or https://plugdata.org/
  Needed to run patches in AAADoc/PureData:
    -Midi to OSC
    -Sound analysis to OSC
- **XnView**
  Image browser with batch and conversion capabilities
  https://www.xnview.com/en/
  MP or Classic
- **Wings3d**
  3d Editor derived from Symbolics/Nendo.
  Nice export tools to .obj files.
  https://www.wings3d.com/
- **Blender**
  The Open Source 3d Editor
  https://www.blender.org/

Install 3/4: AAASeed



- Get **files** from **https://AAASeed.org/files/**
- **Use the latest Installer**

  Install with AAASeed_Setup.1.3.1.exe (2025 April)

  Need a Reboot the first Time

  Download and add libcurl.dll and cpr.dll to AAADll Folder

  No updater for now

  Update with the latest AAASeed.7z
  - **Replace previous AAASeed folder**
  - **You can rename previous and current AAASeed folders keep different versions if you want**

- **Folders**

  -AAAFoundation **AAASeed** stuff and your work are stored there

  -AAADll Dlls associated (External software libraries), the installer define a system path to it

  -AAASeed What The AAASeed team maintain/update/change, executables are there

  -AAAUser Where your preferences are stored

  -APPs_Guest Where you should save your APPs

  -Media Where you put content (videos, images, sounds...) with sub folder by projects

  -Install Associated installers, drivers...
- It is possible to install from a local drive / USB Key with no Network.

  A manual install is possible too

  Copying folder and adding the Dlls path to the User or System path)

Install 4/4: Multiscreen

AAASeed supports **up to 6 screens**

On **desktop**

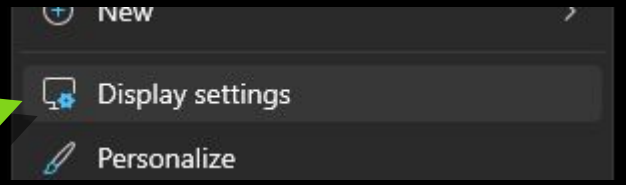
right click menu

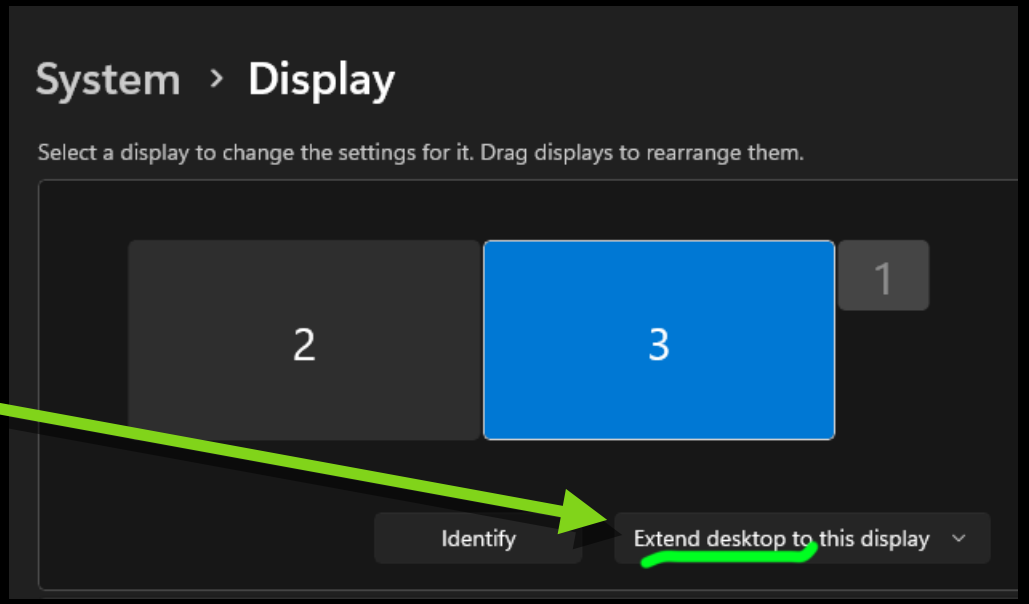
Display Settings item

Dialog

- Set **Extend desktop** mode
- Be careful with the **screen aligment**

  Horizontal and Vertical
- Choose the **right resolution**

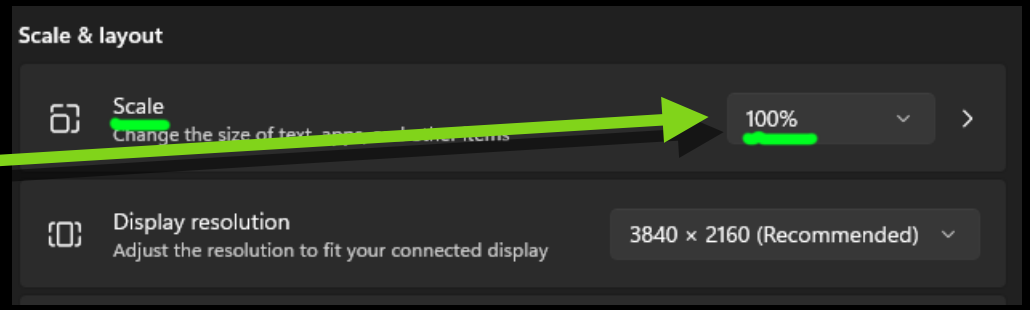




Was mandatory before 2025 May

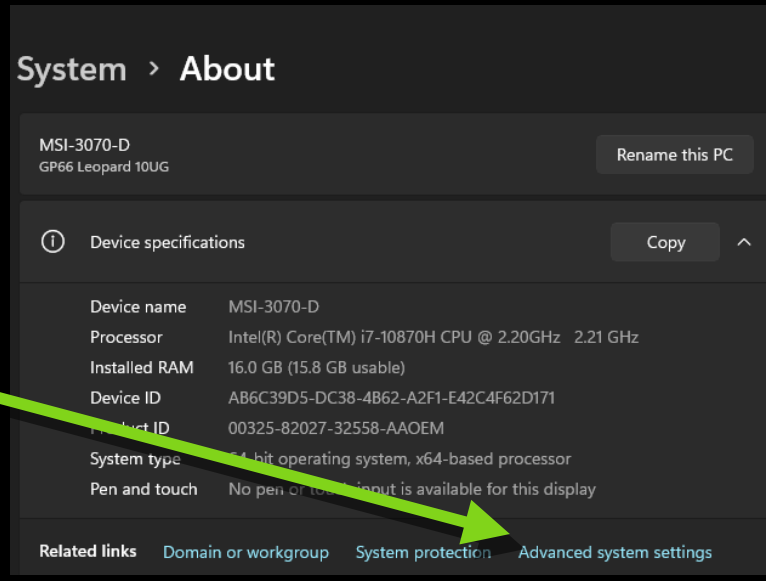
Scale all screens to 100%

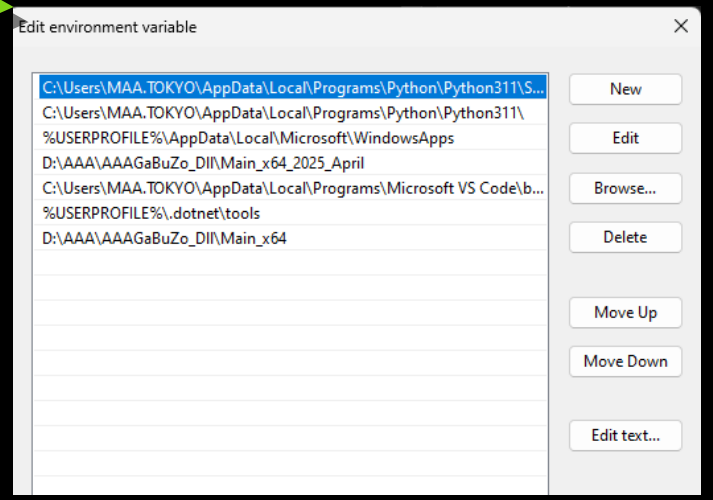
seems ok now, testing



Install TroubleShoot (Archive)

- If **AAASeed** complains about missing dll other that VC … .dll
  - Copy the Dll path from the folder where they are
  - Go to PC / Properties / Advanced system settings
  - Environment Variables
  - Then User variable / Path (User or System) / Edit…
  - New
  - Paste path and eventually change order
  - Validate : Ok then Ok then Ok
  - Launch AAASeed_Metal.exe
- If it still complains about some missing VC … .dll
  - Execute VC_redist_2015_2022 … .exe
- If it still complains about missing VCOMP120.dll
  - Execute vcredist_2013_x64.exe





AAASeed Folders: Inside

AAASeed folder name can be changed

- Users use this to keep different versions

We call it the start folder

- in lua aaa.dir.get_dir_start()

This is what we maintain and update

- As User/Guest don’t store your stuff here
- For developers: this is the Git repository

2 executables

- AAASeed_Metal.exe
- AAASeed_Wood.exe for machine with no AVX2 Extensions
    - AAASeed_Metal.exe just crash at start

3 files

- README.md
    - Detail the install and tree structure
    - Document the start sequence
    - Preview in VSCode
- ReleaseNotes.md
    - Document the evolution from version to version
- Version.txt
    - Corresponds to the current version of the installer

AAASeed Folders: Inside more

- **AAAAPPs** Folder
  - **APP** is like a document

    An App ↔ A **folder** with a file **`default.layerss_param`** inside

    Examples, Tutorials…

    No way to save an APP for the moment: DUPLICATE FOLDER !!!
  - **AAAAPPS maintained by the core team**

- **AAADoc** Folder
  - Far from finished (state in 2025 May)
  - **Lua** API

    lua_aaaseed_draw.lua

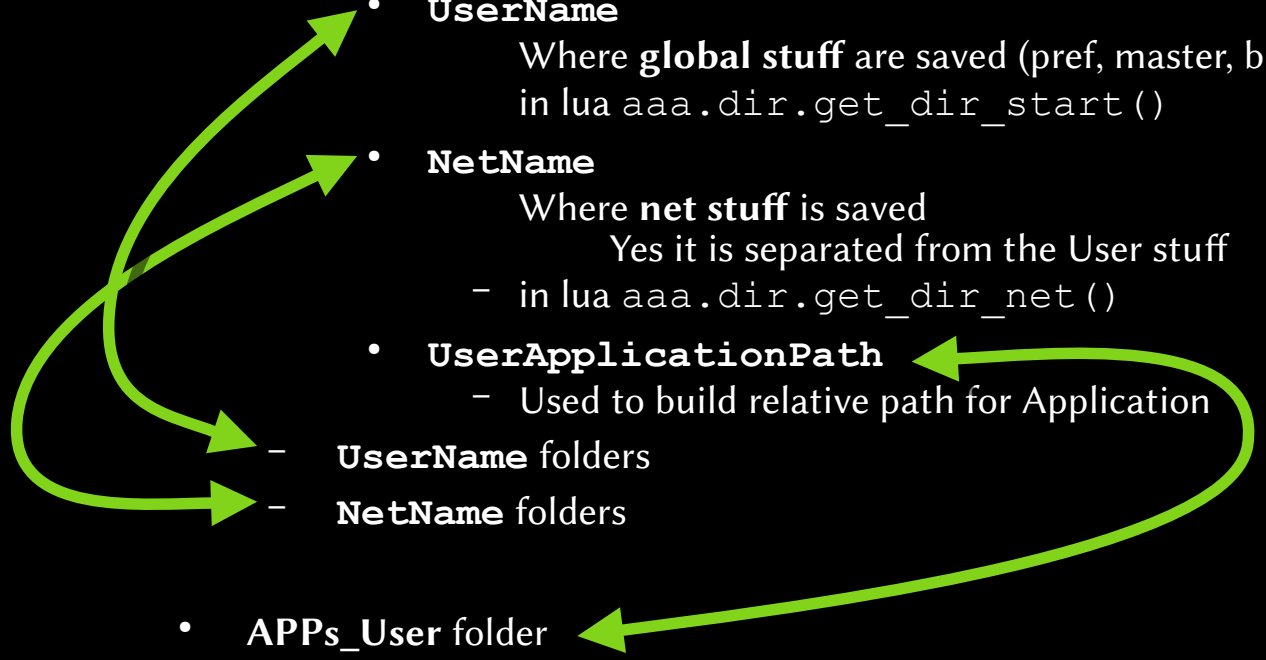
    lua_aaaseed_interface.lua

- **AAAKernel** Folder
  - **Developer space**

    Some resources: lua code, shaders, OpenCl, Fonts, Textures…
  - in lua `aaa.dir.get_dir_kernel()`

AAASeed Folders: going Up

- **AAASeed** the Startup folder
- **AAAUser** folder
  - **`AAAWho.txt`** made of **3 lines:**
    - **`UserName`**
      Where **global stuff** are saved (pref, master, binds...)
      in lua `aaa.dir.get_dir_start()`
    - **`NetName`**
      Where **net stuff** is saved
      Yes it is separated from the User stuff
      - in lua `aaa.dir.get_dir_net()`
    - **`UserApplicationPath`**
      - Used to build relative path for Application
  - **`UserName`** folders
  - **`NetName`** folders
- **APPs_User** folder
  - **APPs_Guest** by default
    - Your APPs, MEU protos, data, shaders.....
- **Media** folder
  - images, videos, fonts, 3d objects...
  - **Big** files are there



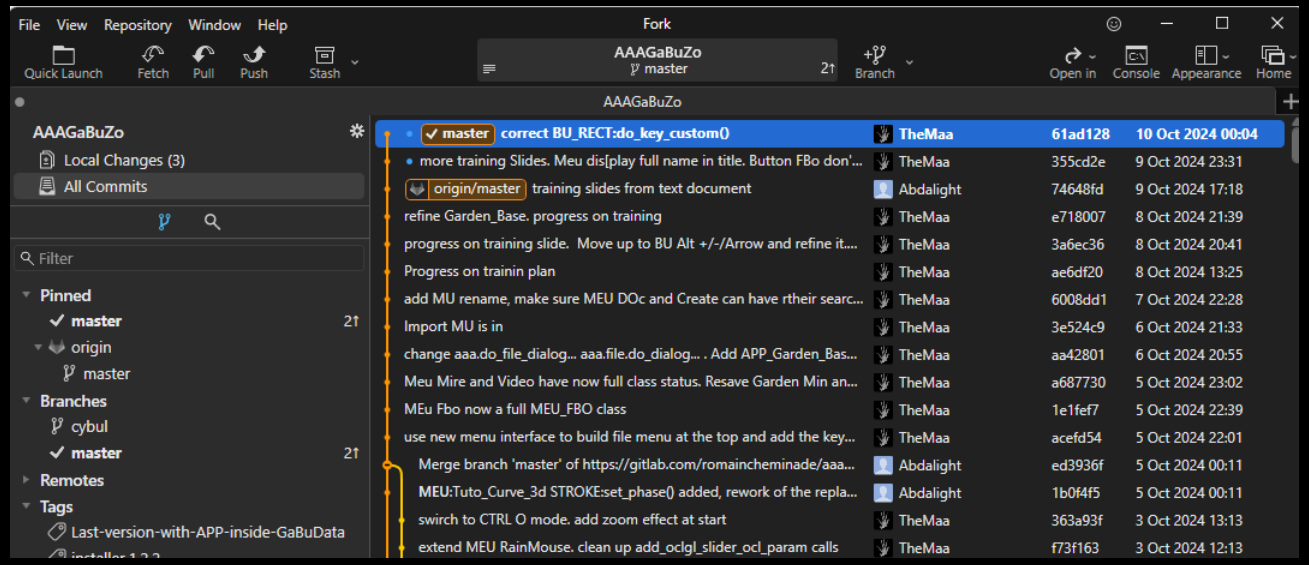
AAASeed Sources

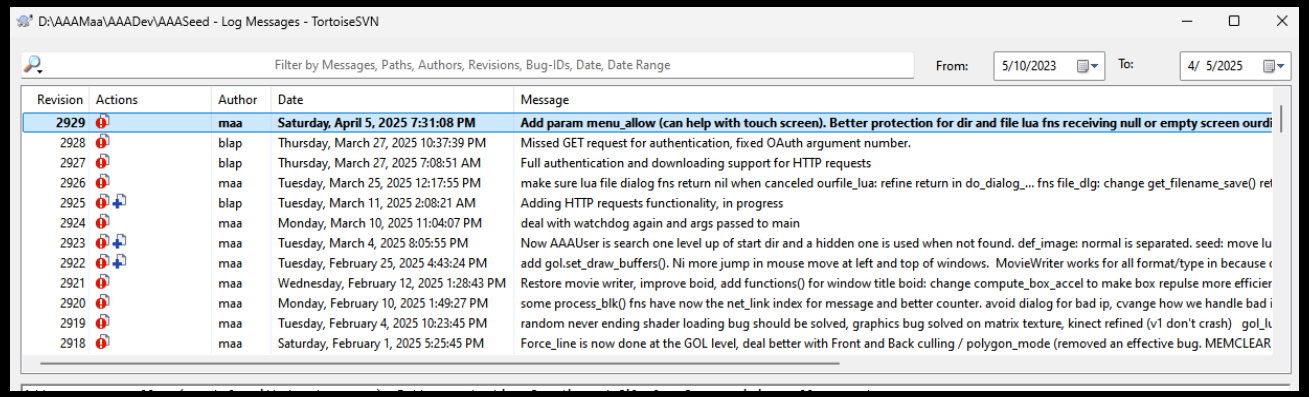
- **AAASeed.org**

  FILES section

- **Developer access**
  git
  -**AAASeed Folder**
  -Lua
  -APPs
  Svn
  -**AAASeed.exe** and dll
  -C++
  -will move to a git

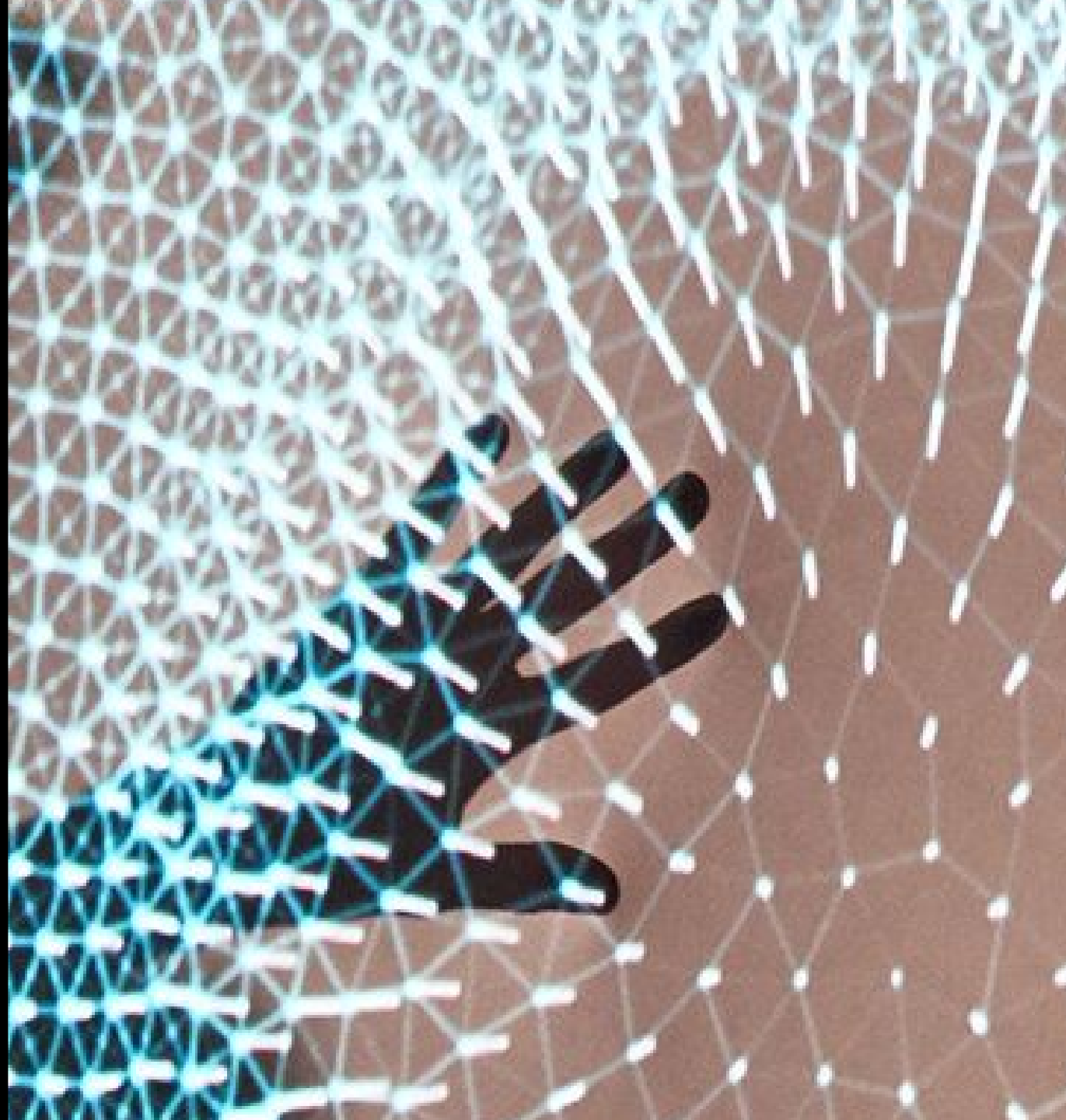
  request Mâa if you want it





AAASeed
An introduction
Part 3:
Baby steps

- **Update AAASeed**
- **First steps**
- **Flatland**
  - Flatland 101
- **Multiscreen**
- **Multiscreen and Flatland**
- **APPlications**

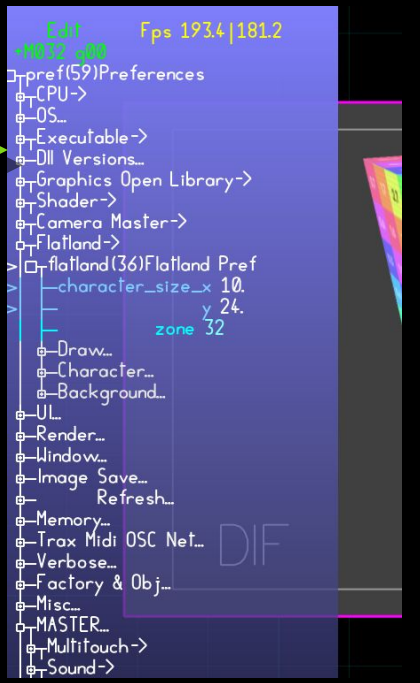


Update AAASeed

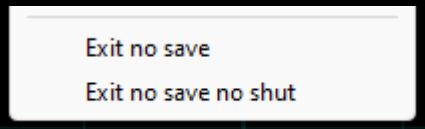
- Get **AAASeed.7z**
  - **AAASeed.org**
    - **FILES** section
- Replace existing AAASeed Folder
  - Reminder
    - **you can rename AAASeed folders**
  - **current version (May 2025) needs 2 more dlls**
    - **libcurl.dll and cpr.dll**
    - Available at AAASeed.org FILES section
    - Put it in AAAFoundation/AAADll

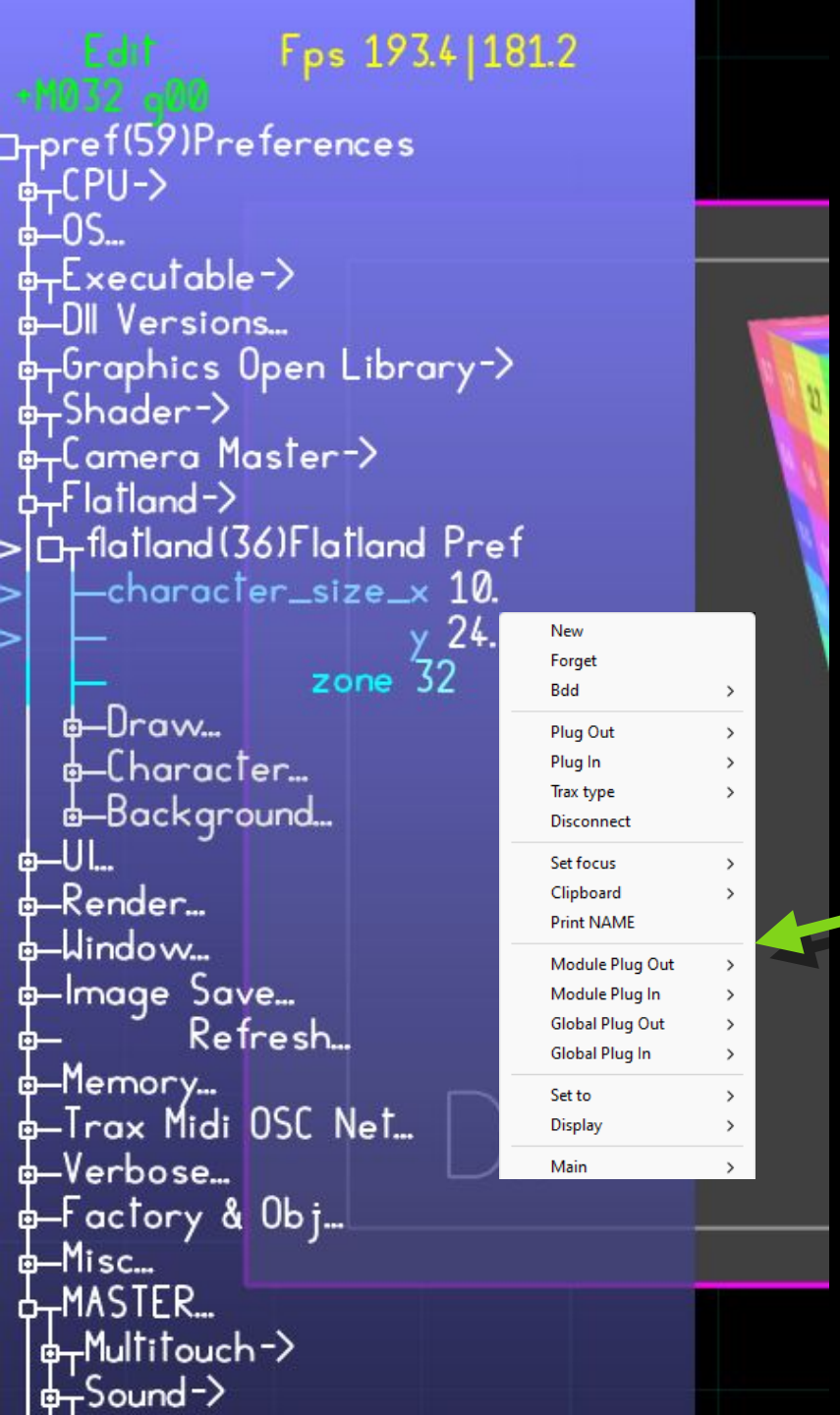
First steps

- **keyboard** first step
  - **w** like (**w**)indow
    switch between window and full screen Module
  - **Tab**
    switch Flatland On and Off
    Ctrl e like **(e)**dit **flip edit mode** (more later on this)
    see feedback at the top left of Flatland
    Red LOCK **Green Edit**
    leave it as **Edit**



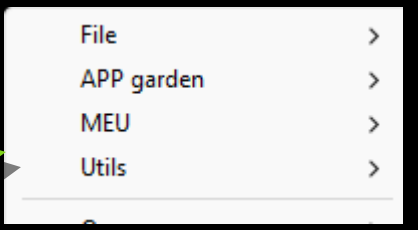
- **Quit, Exit ...**
  - **Double Esc**
    Quit with saving global stuff
  - **Alt F4 or Mouse Right Button/Menu/Exit no save**
    Quit without saving global stuff
    require a confirmation

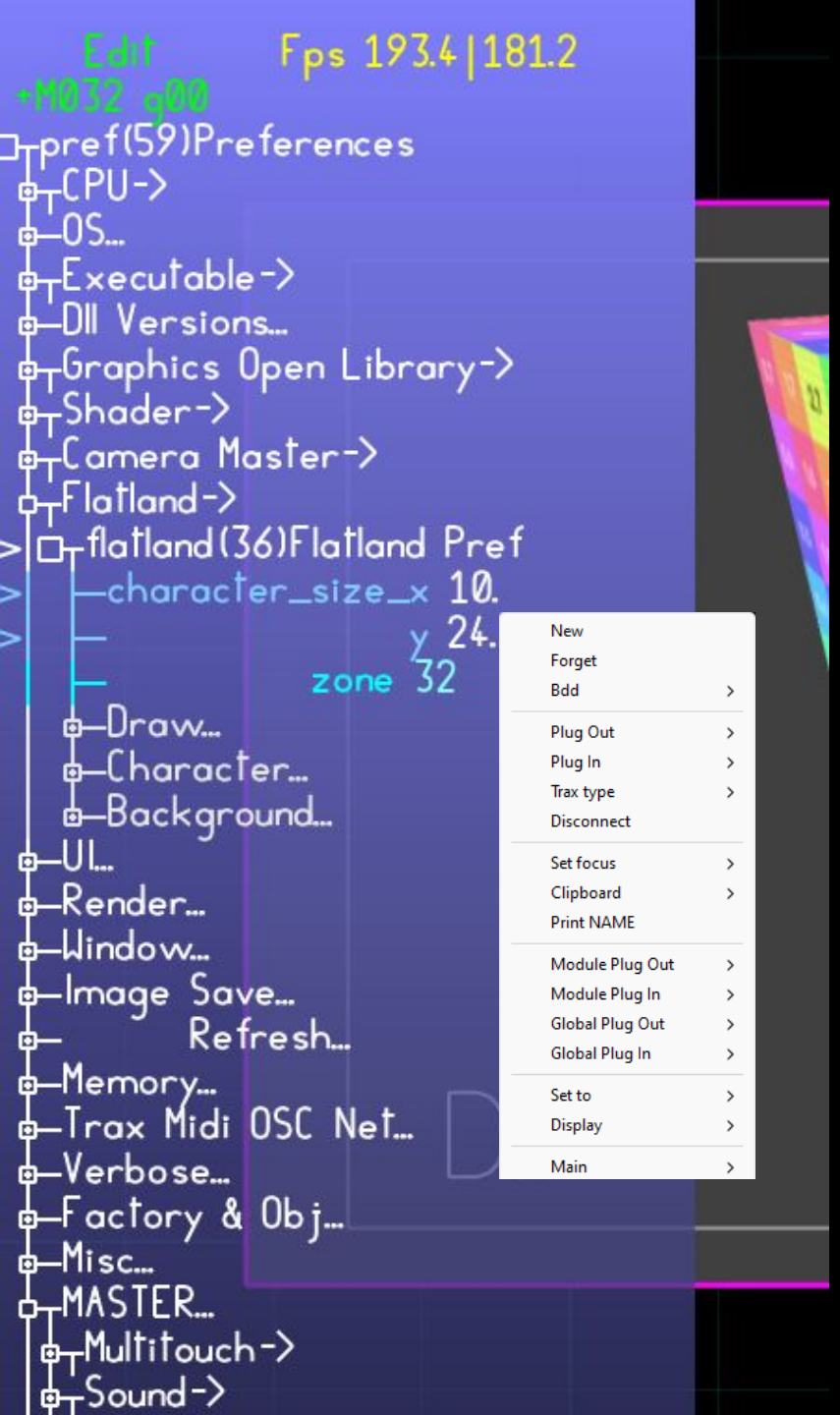




Flatland

- **Tab**
- Original **AAASeed** interface from previous century
  where the **C++ can be seen and used**
  a whole world we will explore later
  made of **c_obj_ui** and **param**
  - **F10 → Preferences**
    example: change values in Flatland (next slide)
  - **Ctrl F10 → Preferences at start**
- **Right Mouse Button → Menus**
  in Flatland → param menu
  out Flatland → main menu
- **GaBuZoMeu**
  Start of Main Menu
- **Flatland have precedence for events and keyboard**
  Needs to be off to use keys with GaBuZoMeu



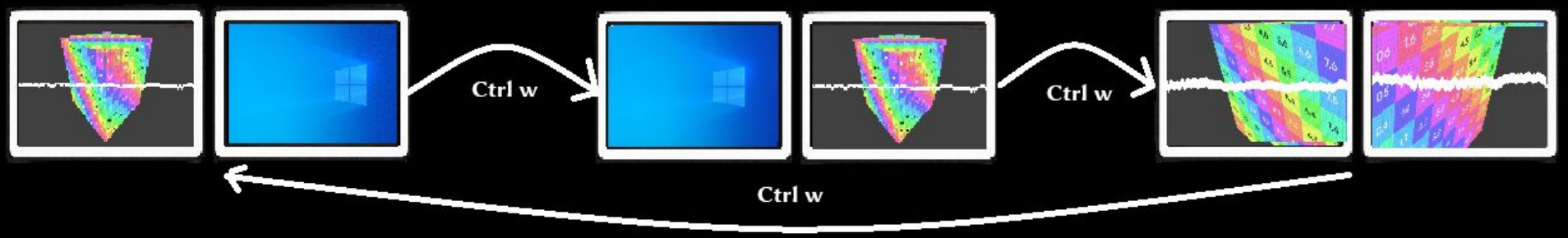


Flatland 101

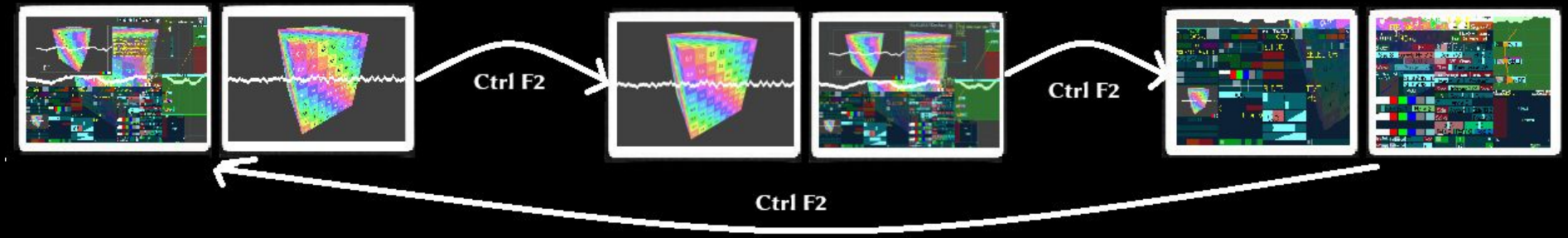
- **Made of params (for parameters)**
- Example: change values character_size_x/y
  - **F10 → Preferences**
  - **Wheel → Scroll**
  - **Click (left Mouse Button)**
    select / open
  - **Click and turn around starting Point**
    change value
  - **Double Click**
    edit dialog
  - **Keys + - * /**
    change value
  - **Ctrl z**
    undo (only last change)
- **Reminder**
  Flatland have precedence for events and keyboard
  Needs to be off to use keys with GaBuZoMeu
  So leave it off for now → **Tab**

AAASeed and Multiscreen

- **w** like (**w**)indow
  switch between **window** and **full scree**n Mode
- **Ctrl w** **circle** through all the possible **full screen configurations**
  Shift **go the reverse way**



- **Ctrl F2** **change UI position on screens**
  Shift **go the reverse way**



- Same principles with more screens (up to 6)

Multiscreen and Flatland

- **Flatland by default on left**

  of AAASeed Window

  which can traverse several screens

- If it is a problem

  switch screen position in desktop

  See previous slide

  or change Flatland position

  Flatland (Tab)

  Preferences (F10)

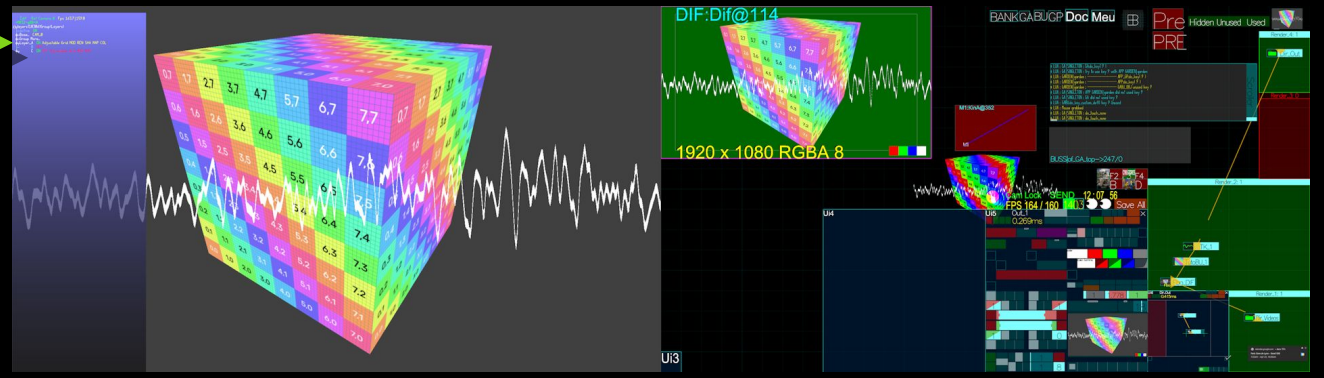
  Flatland/Draw

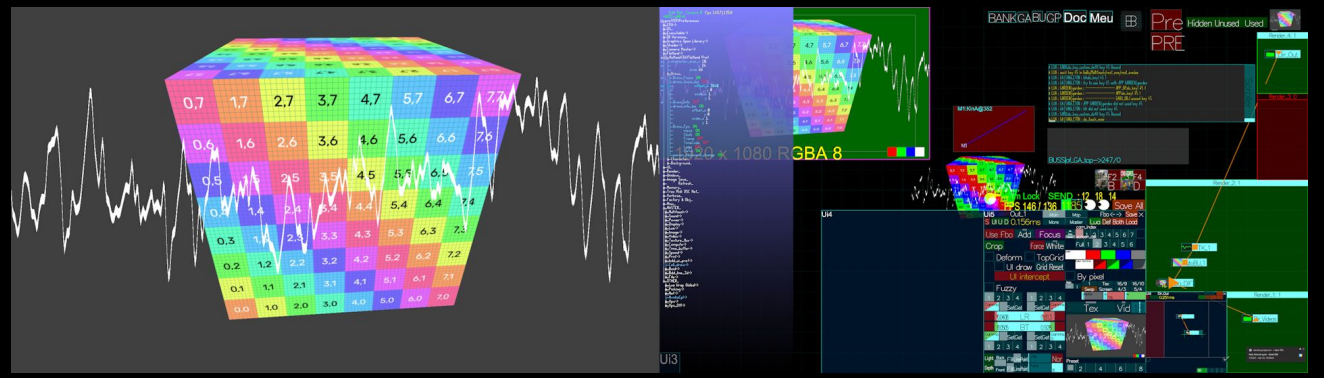
  draw_focus_offset_x/y

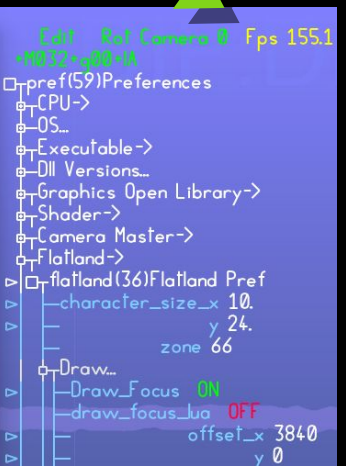
  Careful it can make **Flatland invisible**

  in some configurations

  (**drawn outside**)





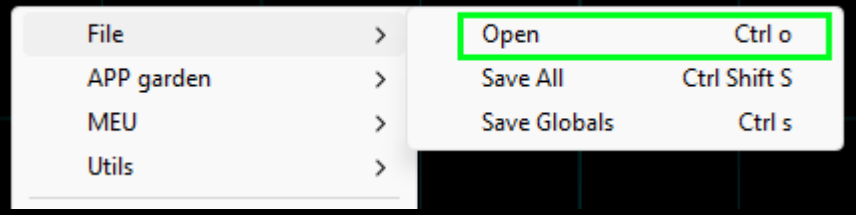


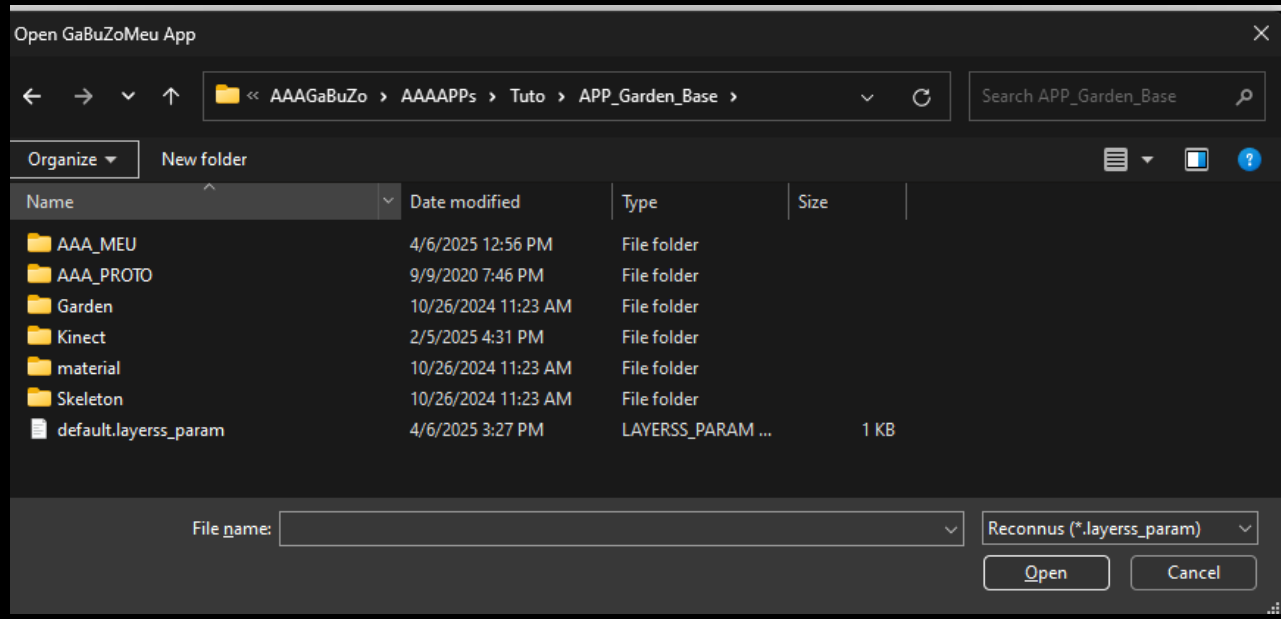
TODO Lua errors

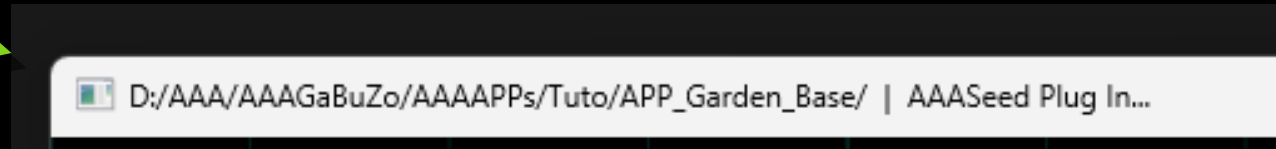
- Dialog
- Menu
- always On to see problems
- Off in permanent installation

APPlications

- Open APP/Application Garden_Base
  - **Use Main Menu File/Open** or **Ctrl o**
- **File Dialog**
  - goto **Start folder**
  - then **AAAApps** folder
  - then **Tuto** folder
  - then **APP_garden_base**
  - **open default.layerss_param**
    The APP is the folder
    The folder is the APP
  - Bug:
    Sometimes you need to open it again
- **Name** in the **window bar**

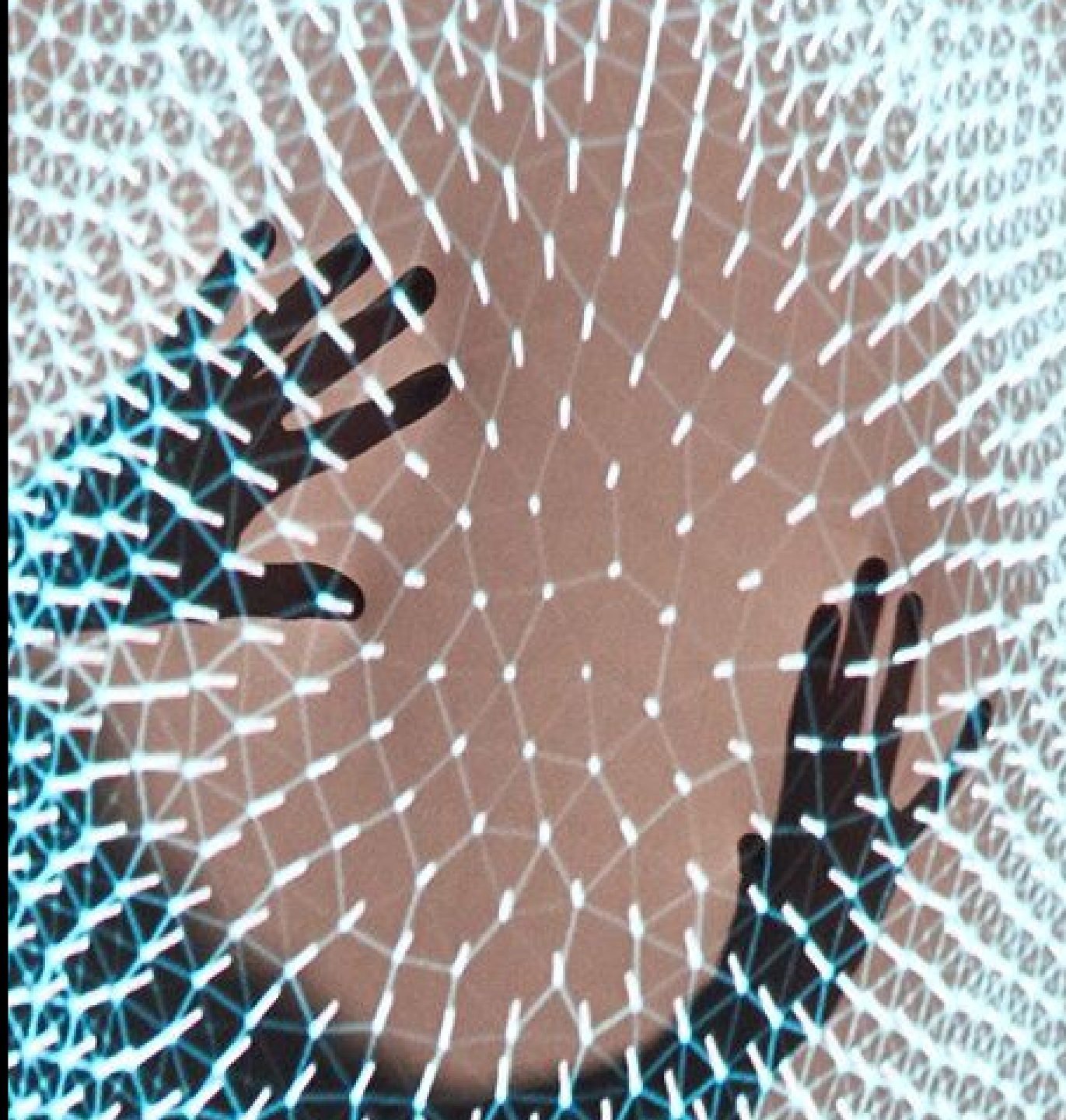






AAASeed
An introduction
Part 4:
BU first contact

- Basic UI
- BU = Box User
- BU Resize Move
- BU StarMenu
- Dialog
- Current BU
- BU Resize more



Basic UI

- **Don't be intimidated**

  need to get used

  UI is efficient.

- **Avoid Caps Lock**
  it can cause confusion

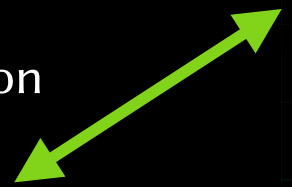
- **F1** or **Ctrl h** like (**h**)elp
  Toggle **help** system

- **F2**
  Toggle **GaBuZoMeu UI**

- **Ctrl F2**
  Force **GaBuZoMeu UI On** and **move it between screens (see Part 3)**

- **F3**
  Toggle **rendering process On/Off**
  BU_Alive visualize its state

  and is **Clickable**



Help the interface is still active except for the keys used by this help :
F1 or CTRL h : Toggle +|− | CTRL Mouse Wheel : Size
Arrows|Page up/Down|Home|End|Mouse Wheel : Navigate

AVOID AVOID AVOID Caps Lock : it will get you lost

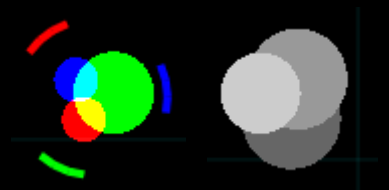
Use a Mouse or TouchScreen

GaBuZoMeu

GA	Global Action
BU	Box User : the interface elements you can interact with
ZO	unused for now (don't worry we working very very hard on it)
MEU	Module Editable Unit : an encapsulated functional Block
MU	Module Unit : a compact MEU view used to control the execution order

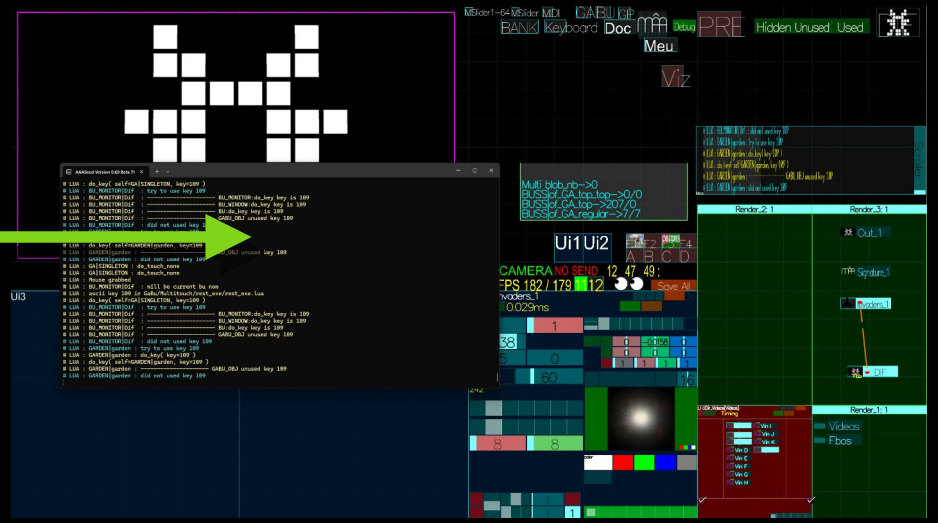
MOUSE

Left Button	Action
Wheel Button	Scroll
Right Button	Menu

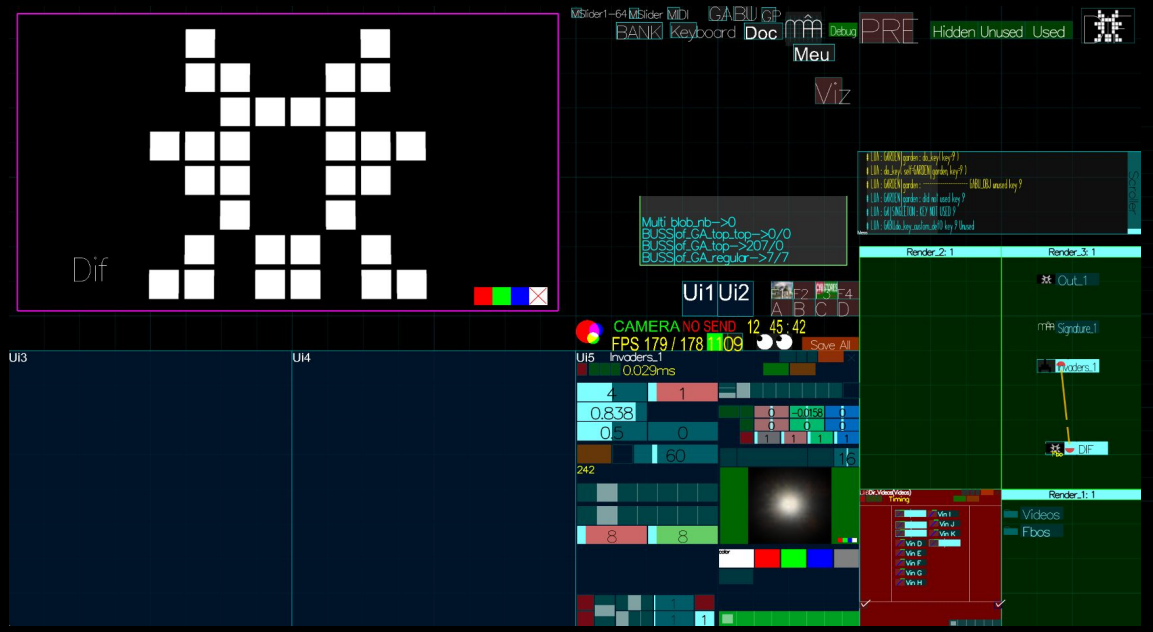


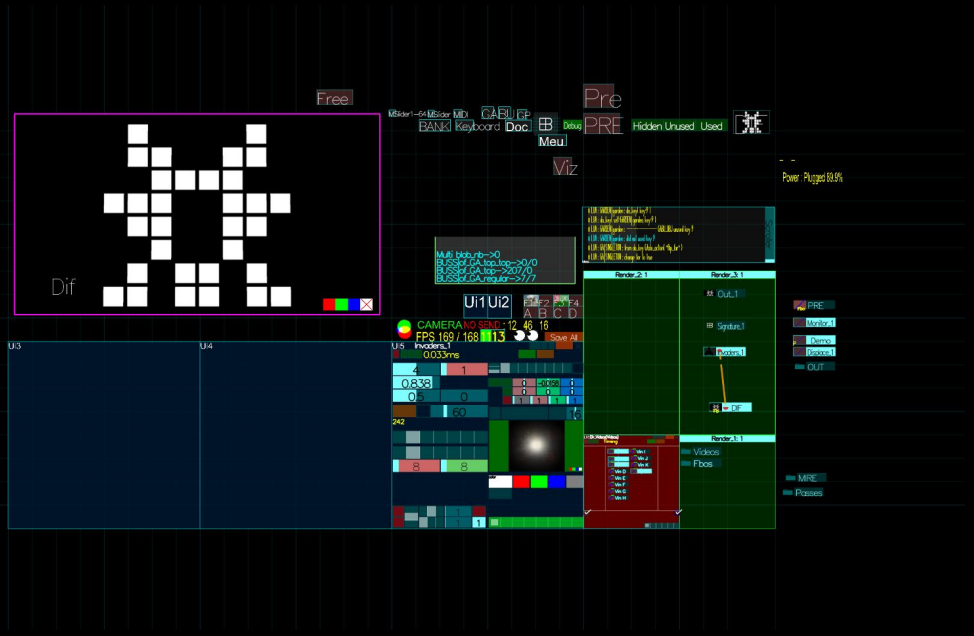
Basic UI more

- **m** like (**m**)essage
  pop the **terminal window**



- **Ctrl Tab**
  Toggle between two **UI Scales**





BU = Box User

- The **atomic UI element** in **AAASeed**.

  Hierarchical:

  BUs contain BUS

  BUS contain BUs
  BU / BUS / BU / BUS ...

  Interactive:

  Click, resize, move, or nest them to build UIs.

- To see this

  Click on the icon of TutoBU_1

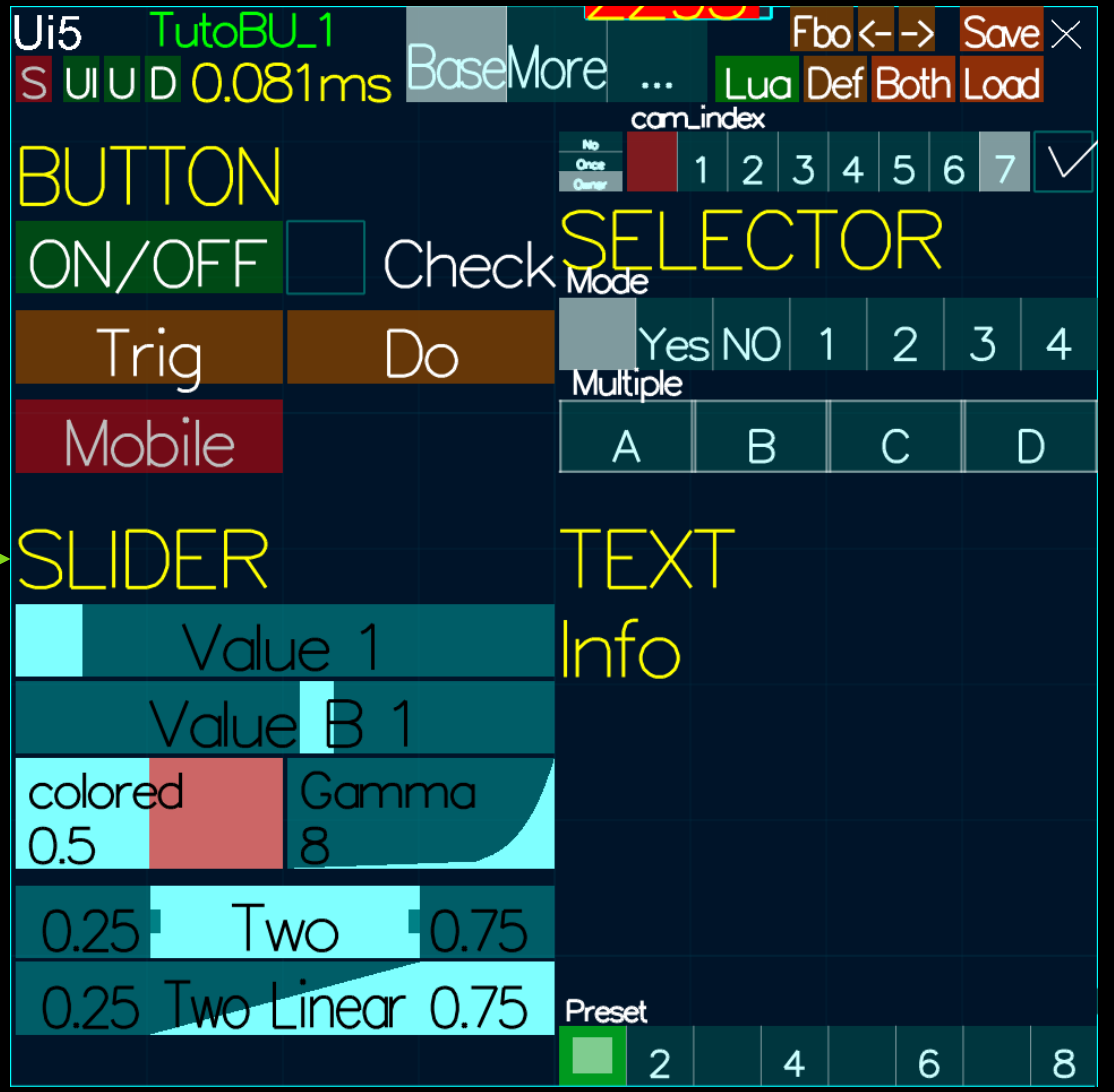
- **Click**
  - Done with **left button**
  - **4 types**

    Simple

    Double

    Triple (rare)

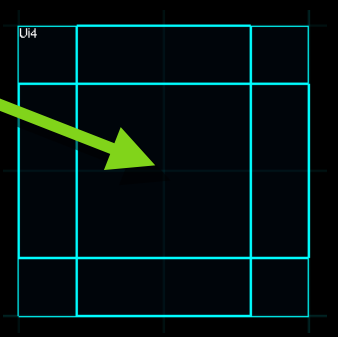
    Long (stay fix with button pressed)



BU Resize Move

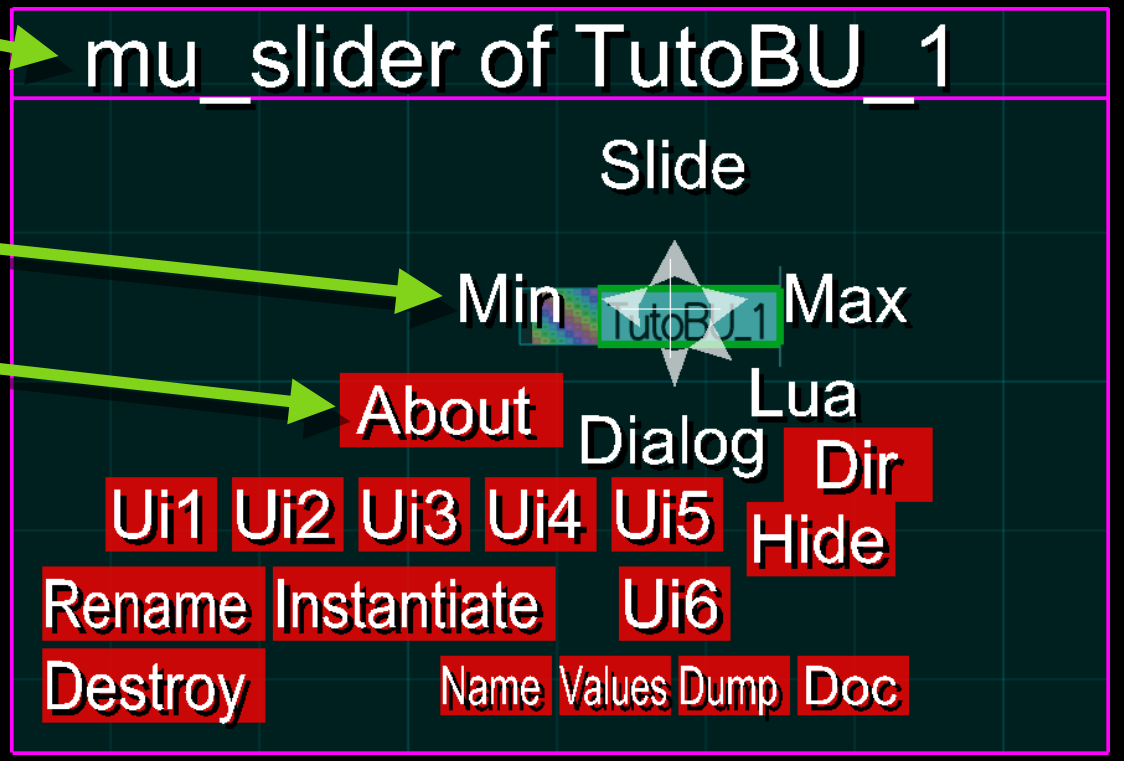
- **Hold Alt**
  - **Transparent**
    - You can see under
  - **Blue Flashing Frame**

- **Click with Alt**
  - **Move**
    - **Center zone**
  - **Resize**
    - 8 **edge zones**



BU StarMenu

- **Example:**
  Click on TutoBU_1 slide
  keeping button pushed
- **StarMenu**
  Name of its **BU** at the **top**
  Also named **UIF** for **UI Fast**
  AAASeed proximity Ui
  It is a **Contextual Menu** with
  - **Zones** (White texts)
    go in the arrow direction
  - **Button** (Red rectangles)
    move on the button
- **Quick Drag-out**
  Not all BU display their **StarMenu on Click**
  They **nearly all have one**
  In this case, click, and with the **mouse button still pressed, go quickly out of the BU**
  rectangle area: StarMenu Appears

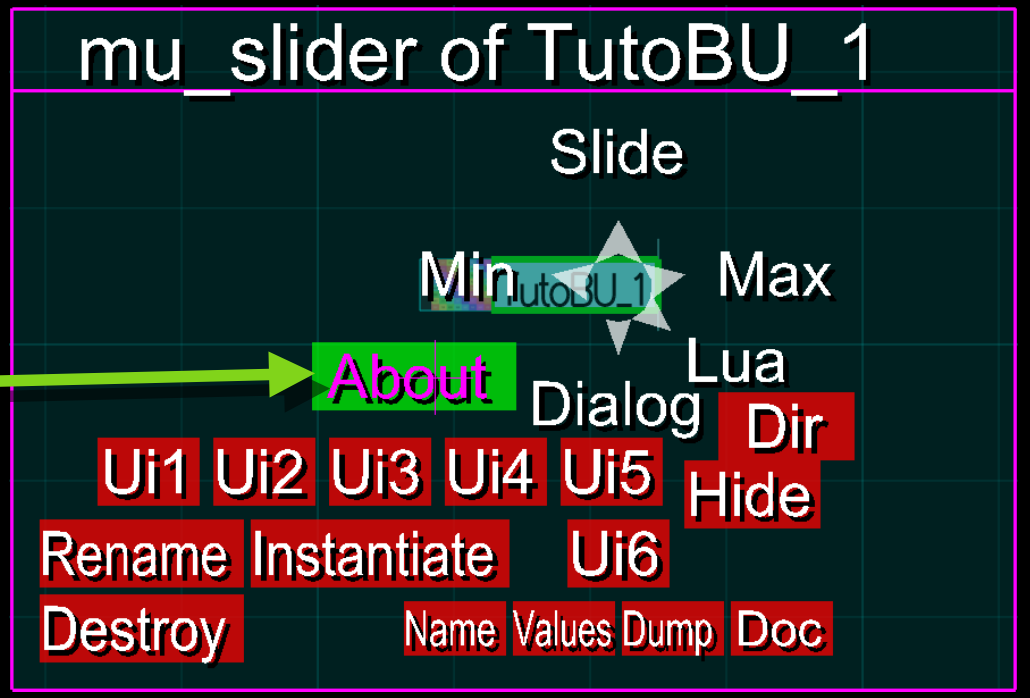


Dialog

- Example:

  Click on TutoBU_1 slider

  use **About** button



- **Dialog**

  Flashing Red for attention

  Move / Resize using Alt as Always

  Click to dismiss

Current BU

- **Clicked** BU

  becomes **current**

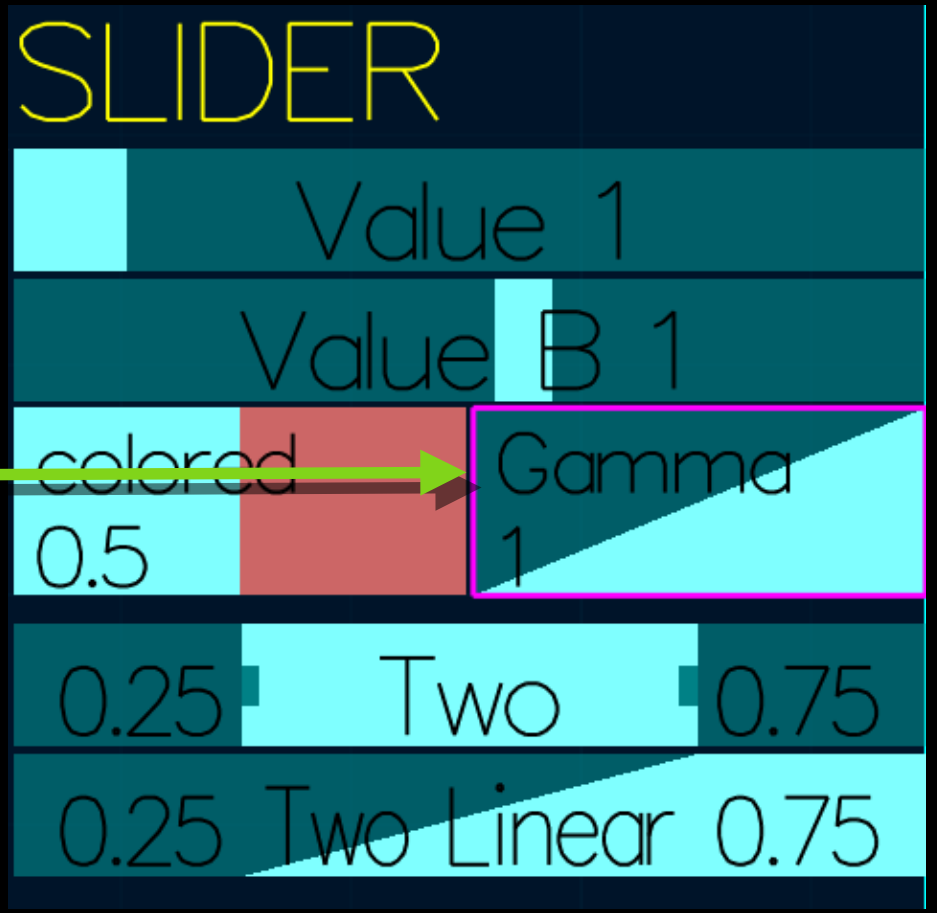
  Violet frame

  Receives keyboard input

- **Arrows key to navigate**

  perhaps some bugs left

  we need feedback (2025 July)



BU Resize more

- **Alt click** move resize
  - **Alt Arrows → Move**
  - **Alt + - * / → Size**
- 3 states:

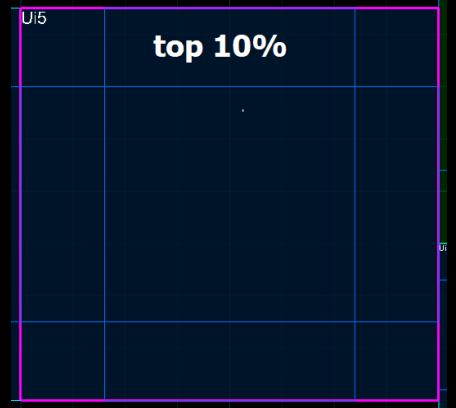
  Normal / Full / Mini
- **Shortcuts**

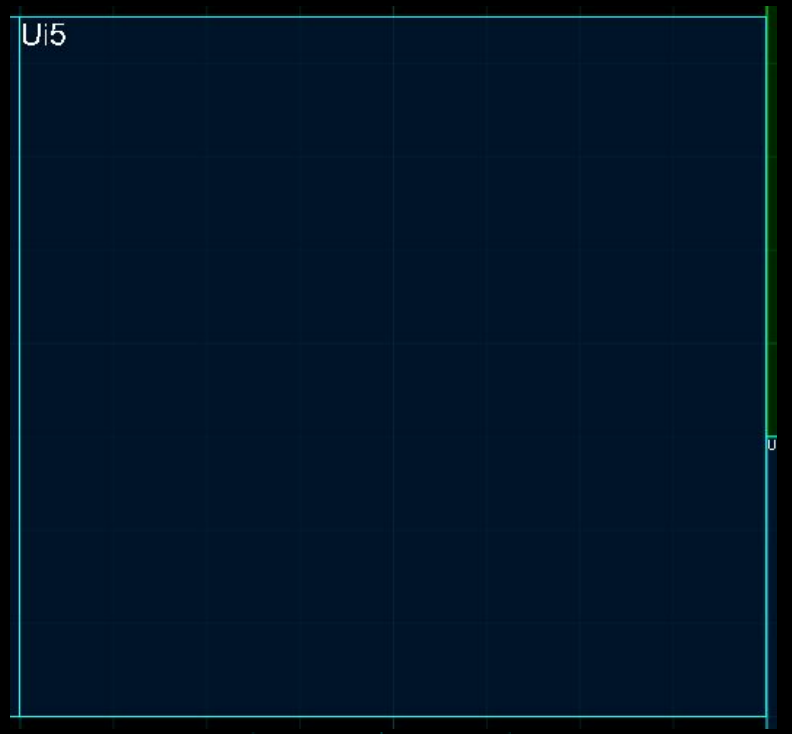
  Double-click

  Space Toggle **Normal ↔ Full**

  Double-click top 10% → Mini

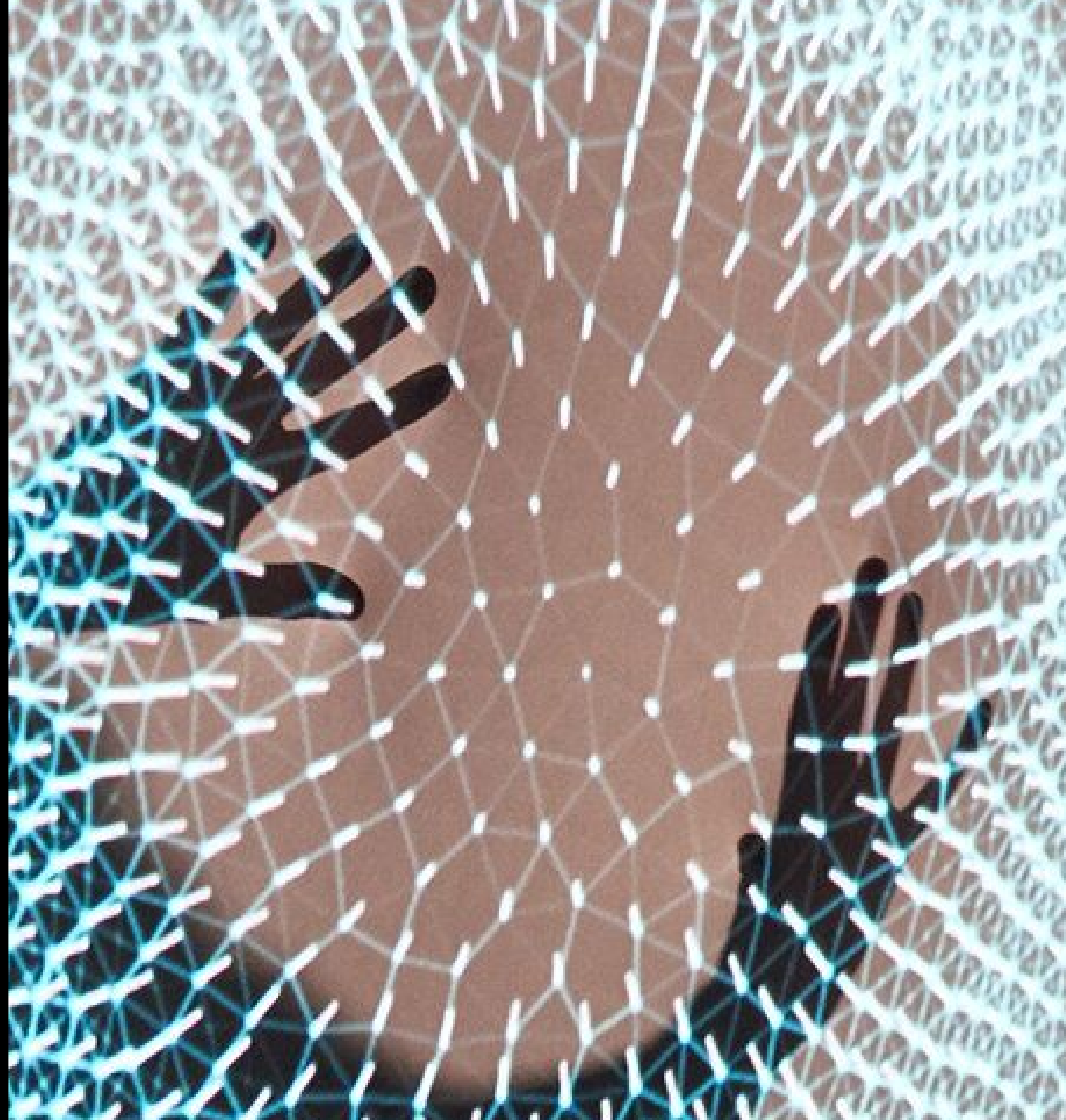
  Ctrl Space Toggle **Normal ↔ Mini**





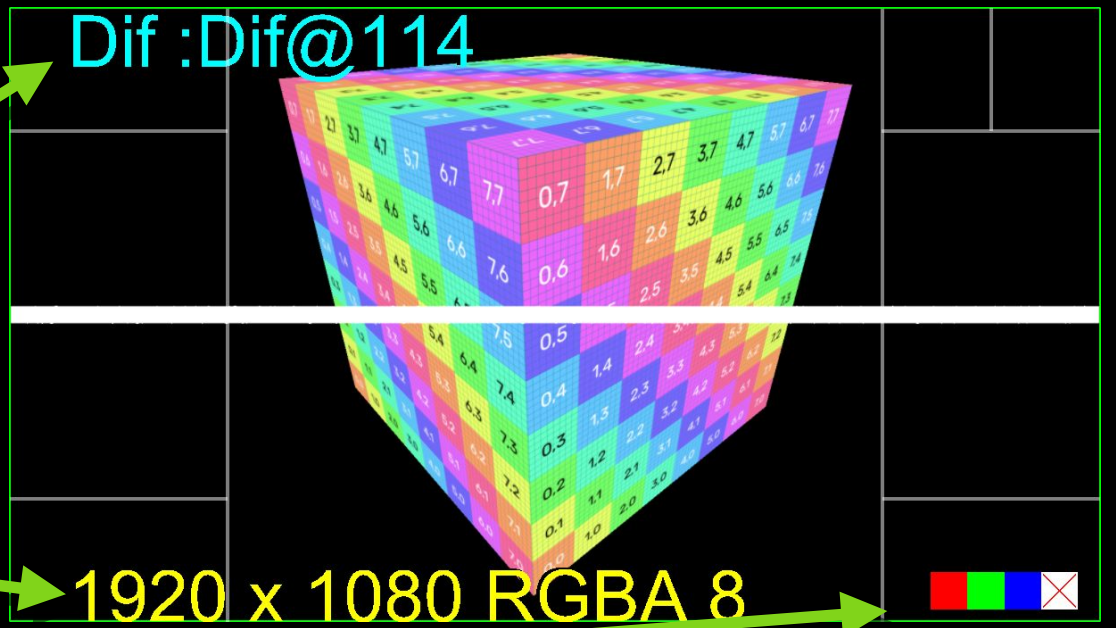
AAASeed
An introduction
Part 5:
BU_MONITOR

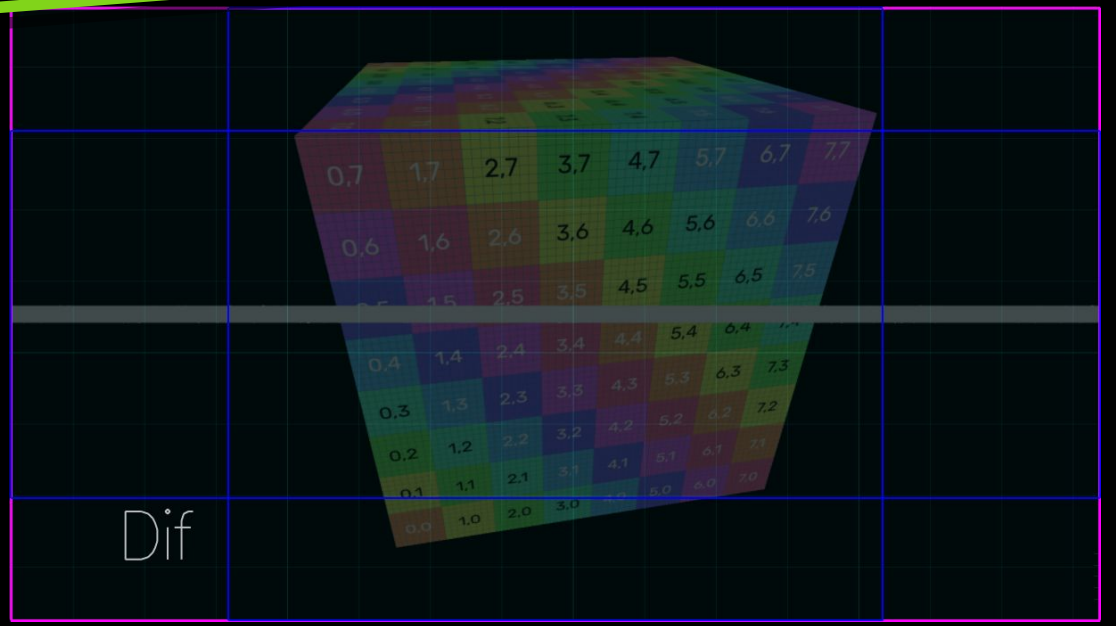
- Channels
- Alpha
- Click Zones
- StarMenu

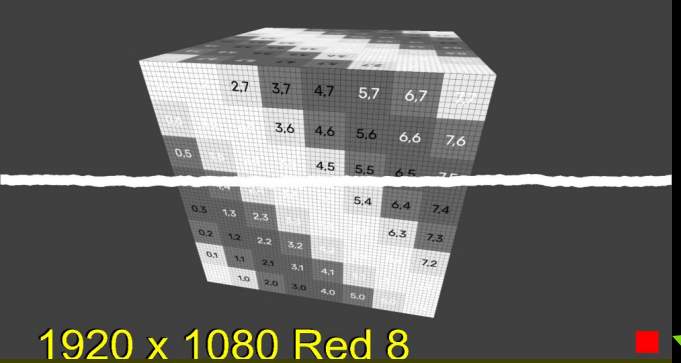


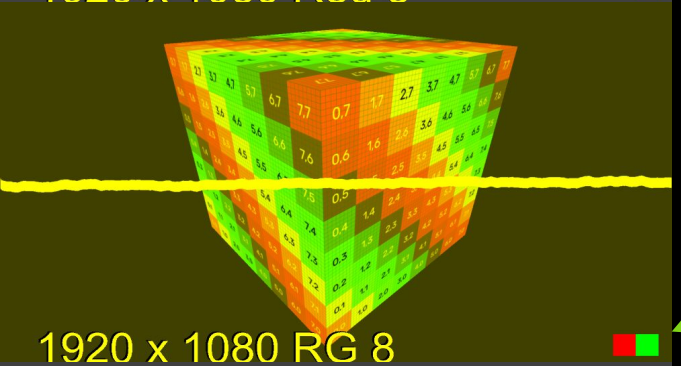
BU_MONITOR

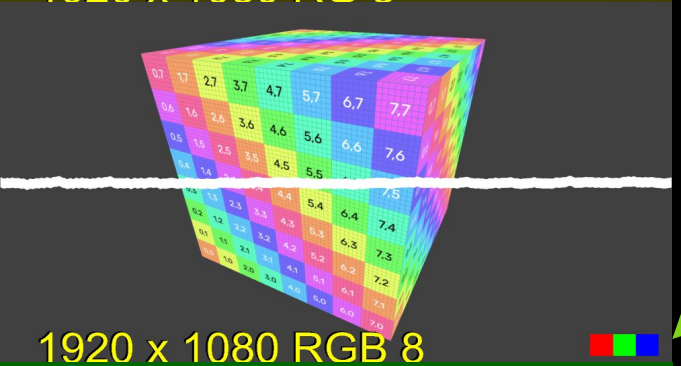
- Display Image
- When clicked
  - Display information on image
    - cyan text at top left
      - which texture
    - yellow text at bottom left
      - texture format
    - RGBA at bottom right
      - channels and Alpha mode
  - Display Click Zones
    - the white rectangles
      - See Click Zones on next slides
- When click is released
  - depending where (in which click zone)
    - a command can be executed
- Move/Resize as usual
  - Alt...

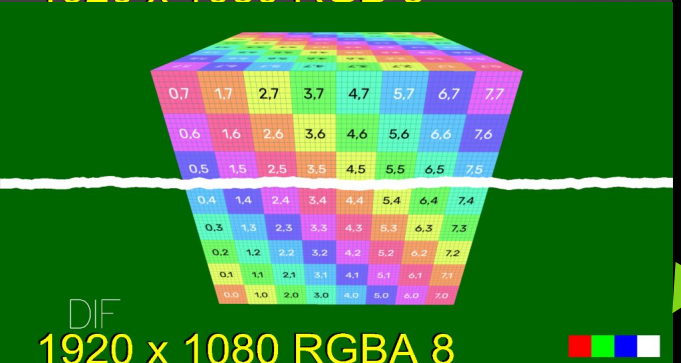










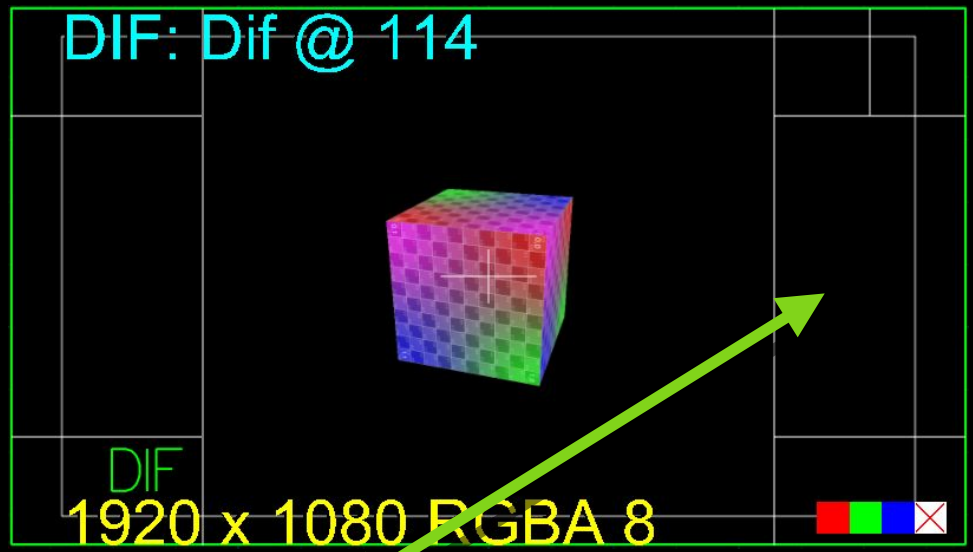


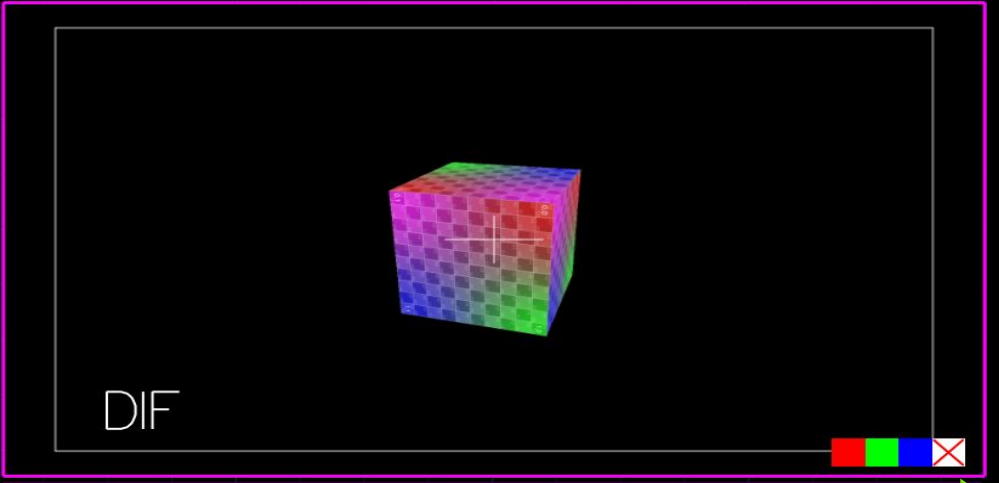
BU_MONITOR Channels

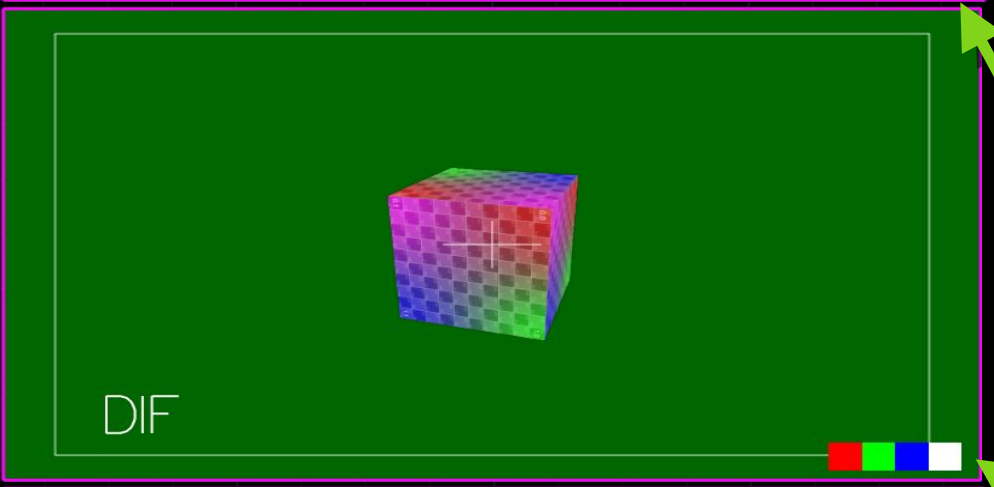
- In fact BU_MONITOR **shows textures**
- **Textures have a number of channels**
  - **1 or R for Red**
    showed as **greyscale**
  - **2 or RG for Red + Green**
    - showed as **yellow**
      no Blue
  - **3 or RGB for Red + Green + Blue**
    full color
  - 4 or **RGBA for RGB + Alpha**
    like RGB but with transparency/Opacity
    0 mean transparent, 1 Opaque
    different method to display Alpha
    Regular / Inverse / No
    See next slide

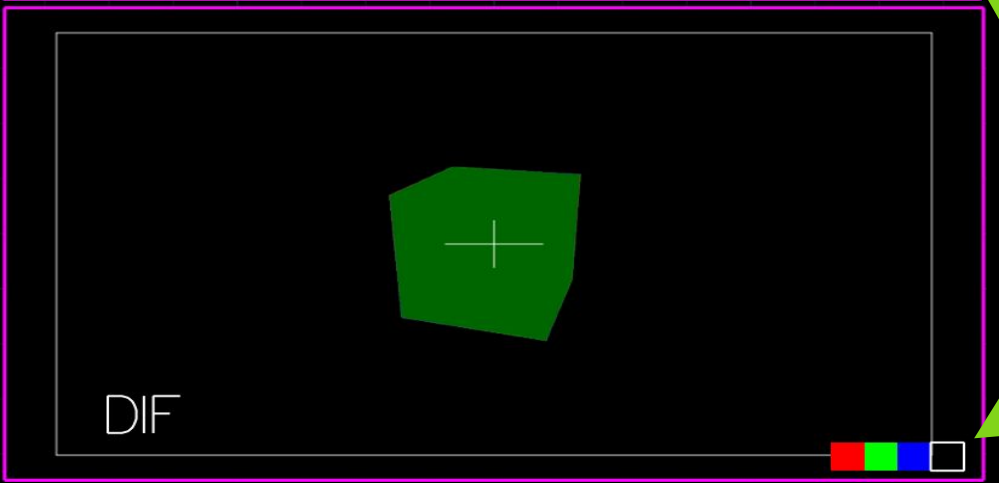
BU_MONITOR Alpha

- **Click zones**









- **Click in Right Middle**

  change how

  alpha is displayed (only RGBA)

  same as **Key + / -**

BU_MONITOR Click Zones

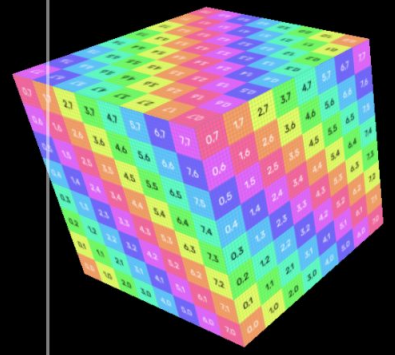
When the click is released the command is selected and executed

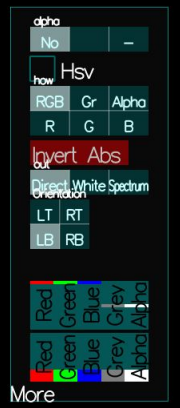
Flip Bank/Bind information

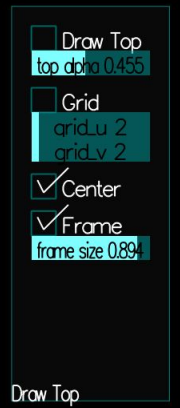
Flip Window More

Flip Window Draw Top

DIF:Dif@114







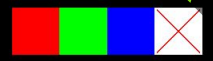
Flip both information

Flip Alpha Mode or Key + -

DIF

1920 x 1080 RGBA 8

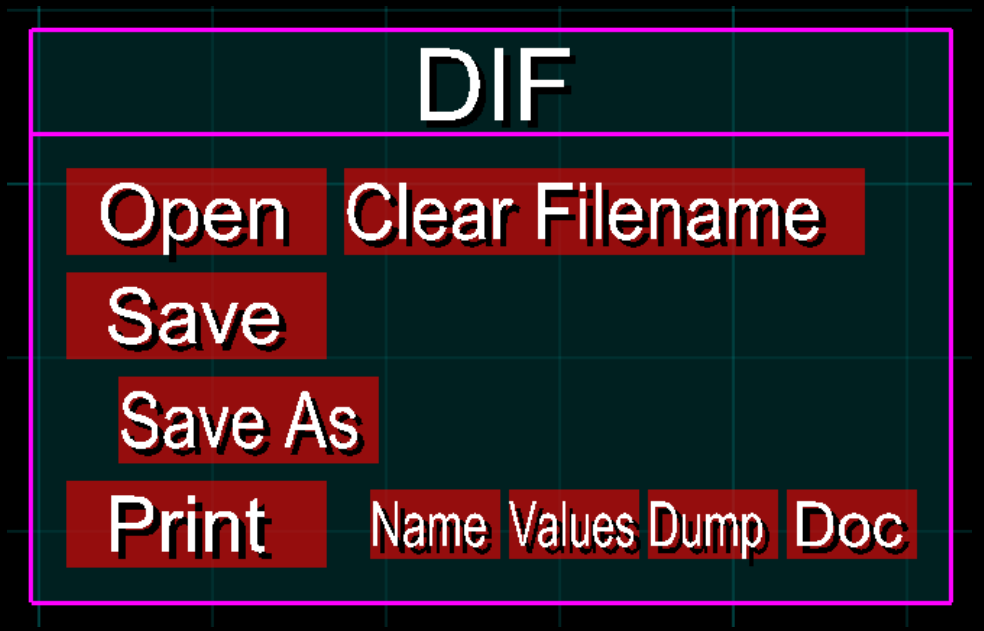
Flip Texture information: resolution and format



Flip channel display

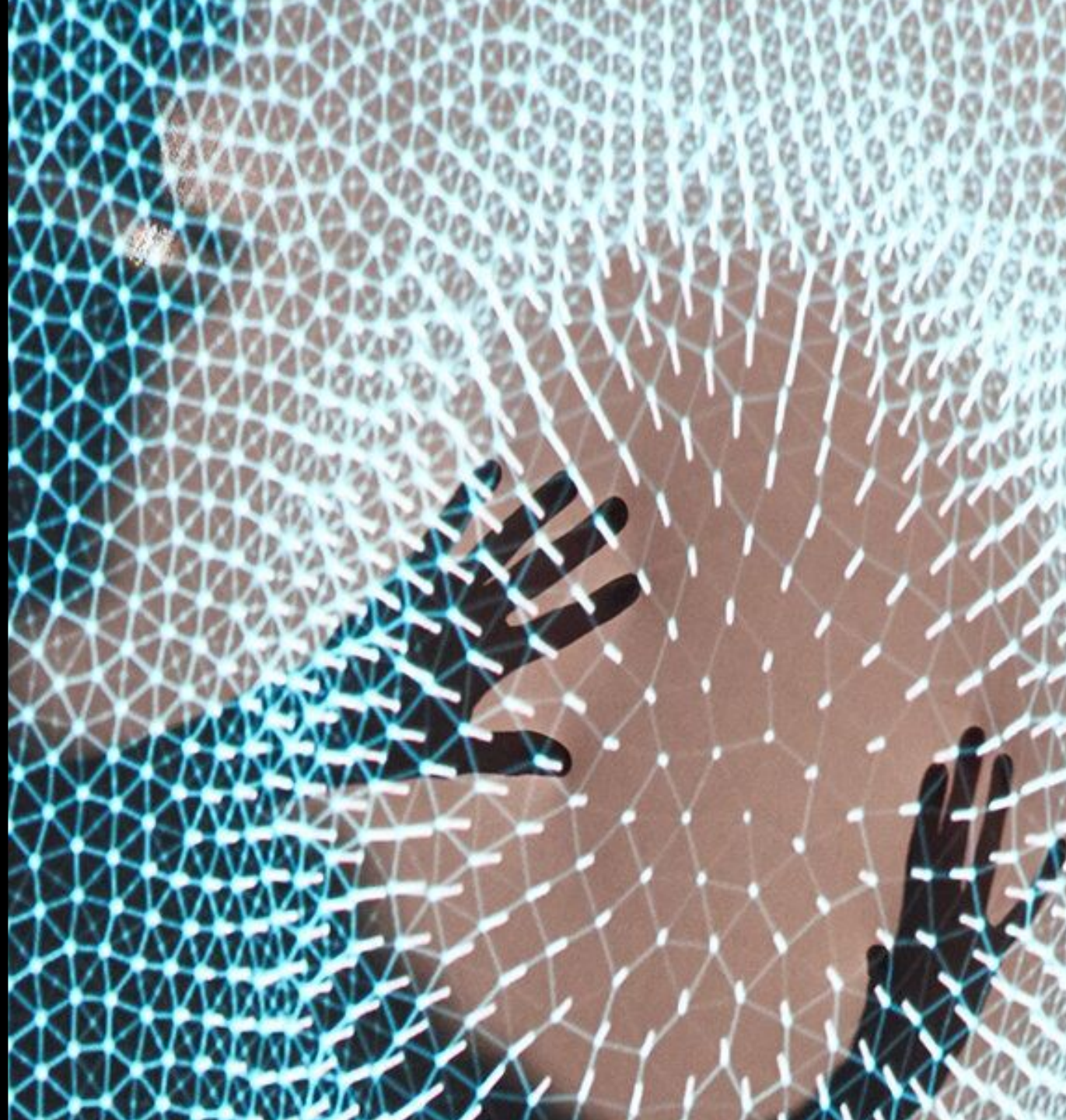
BU_MONITOR StarMenu

- **Use Quick drag-out** like any BU where it doesn't appear directly (defined in Part 4)
- **Open**
  - **Load an image via a file dialog**
    - **at the current bind**
      - later on this (Part 14: Texture)
    - **can choose several files**
      - loaded in successive binds
  - **Set the the filename for this bind**
- **Clear Filename**
  - **Clear the the filename for this bind**
- **Save**
  - **Save the Image to** Media/AAASnapshots
  - How it is saved in Flatland Prefs/Image Save
  - **Note that Save do not use the bind filename set by Open**
- **Save As**
  - **Save with a file dialog to choose location**
- **Print**
  - **Print the image**
  - How it is printed in Flatland Prefs/MASTER/Print



AAASeed
An introduction
Part 6:
BU is all you need

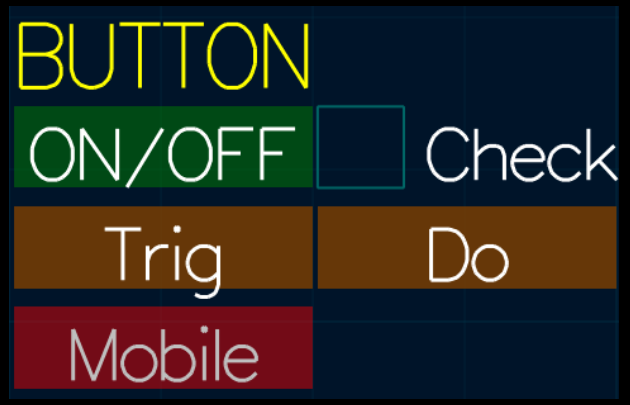
- Basic BU
- BUTTON
- SLIDER
- SLIDER_TWO
- SELECTOR
- BU_TEXT

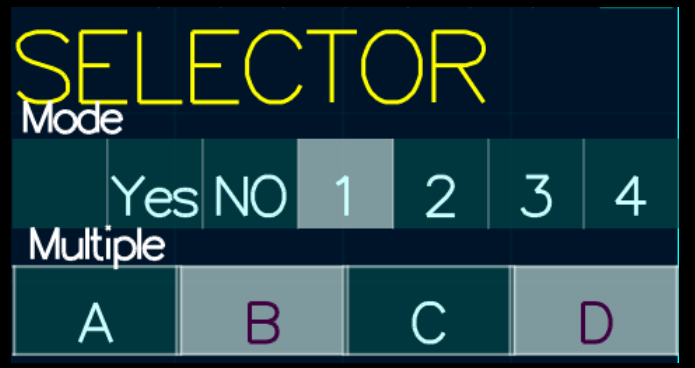


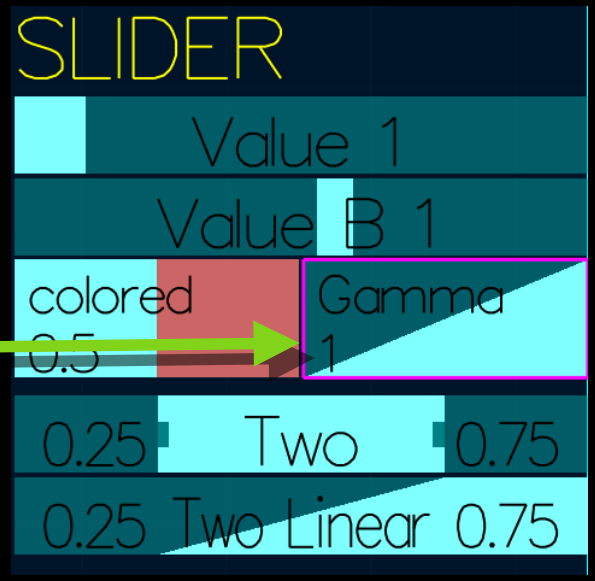
Basic BU

- Click on Train_BU icon

BUTTON
SLIDER
SELECTOR
BU_TEXT







- Current **BU** (reminder)
  receives keyboard
  violet rectangle
  navigate with **arrows keys**

BUTTON

On/Off

2 versions

green / Red

checkbox

Developer note: boolean or integer

Trigger

Launch an Action

Usually Orange

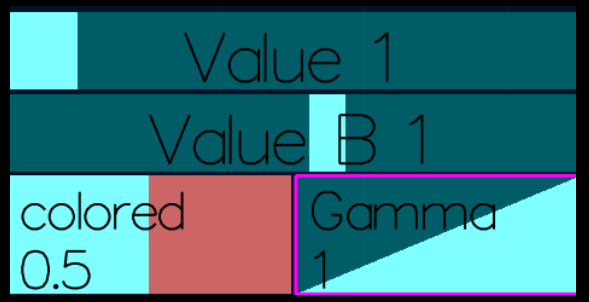
Eventually with **confirmation**

Could be **movable** too (**Alt**)

StarMenu Access

- **Quick drag-out**

SLIDER



- StarMenu
- Slide

  Go up in Slide direction

  Go to external circle

    External circle→ change

    Internal circle → no change

    Intermediate circle → interpolate

  Angle change value

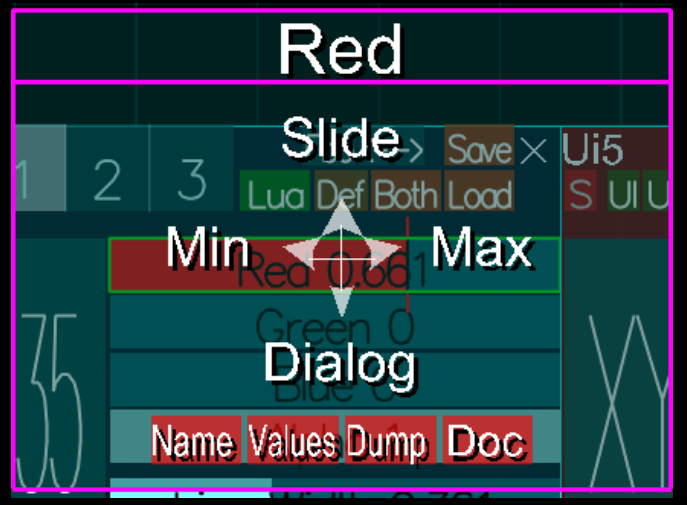
    the more away from center the more precise

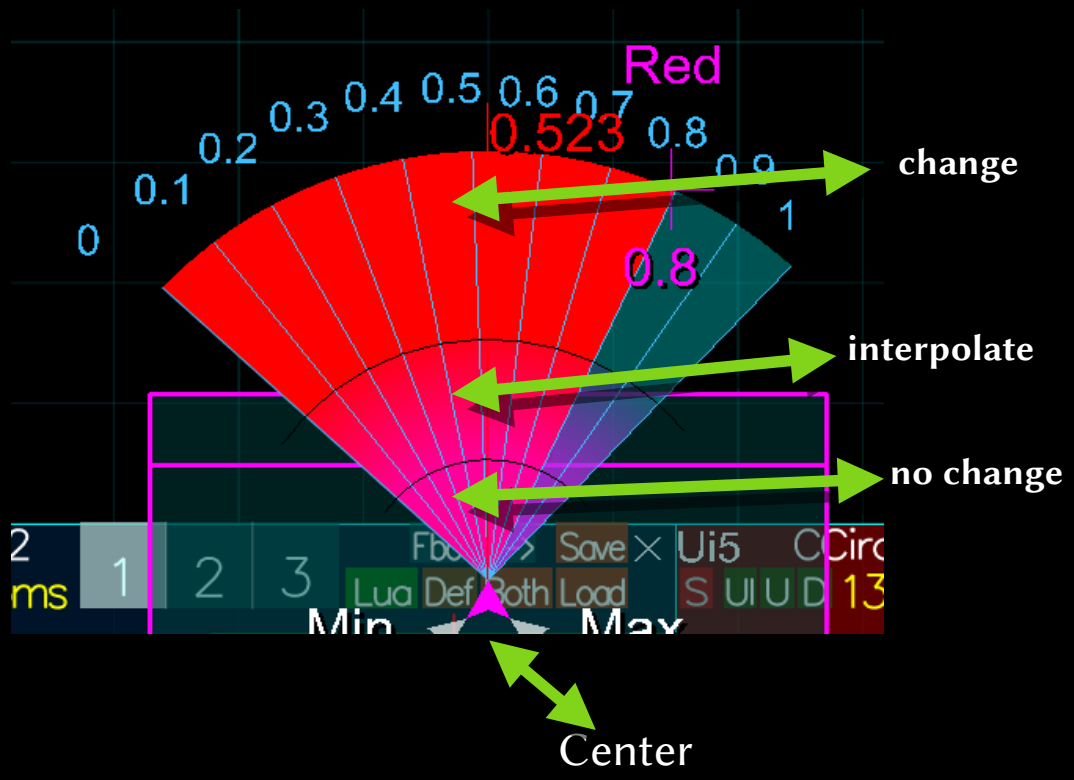
  Go back to internal circle to cancel

  Shift → lock on closest default value

  Ctrl → smaller change from start value

  Ctrl Alt → even smaller change





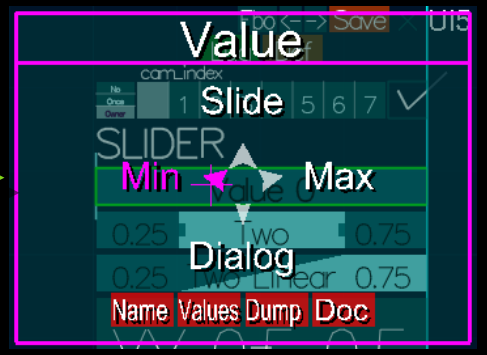
SLIDER

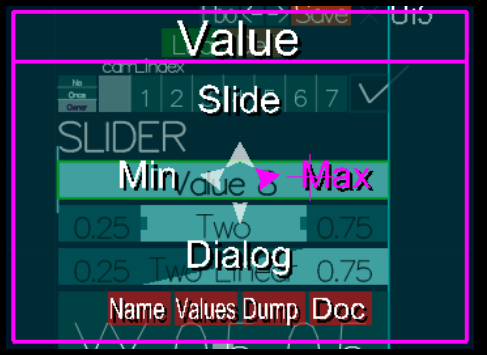
- **Min / Max**
  defined for each slider
  clamping is optional
    constrained by Min/Max bounds
    not by default

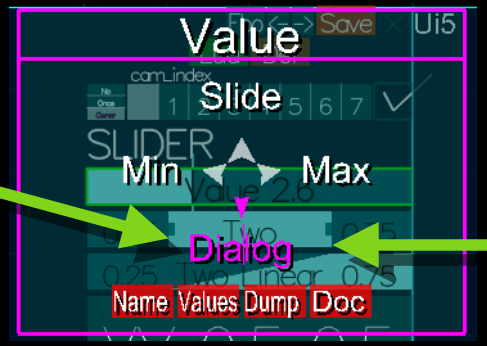
- **Dialog edit**
  Double Click or Dialog zone

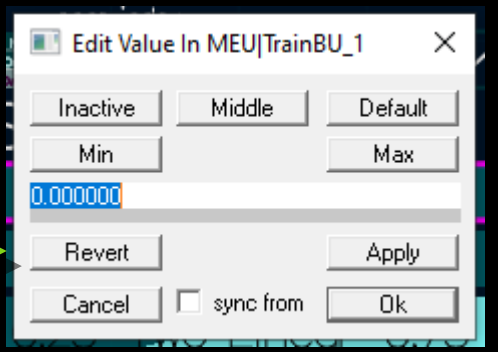
- **Keys**
  act on current BU if no Flatland
    + - * / → change
    Enter → inverse
    . → floor integer (round to inferior integer)
    Home → Default
    End → Inactive
    PageUp → Maximum
    PageDown → Minimum

- Can be **Floating point or integer**

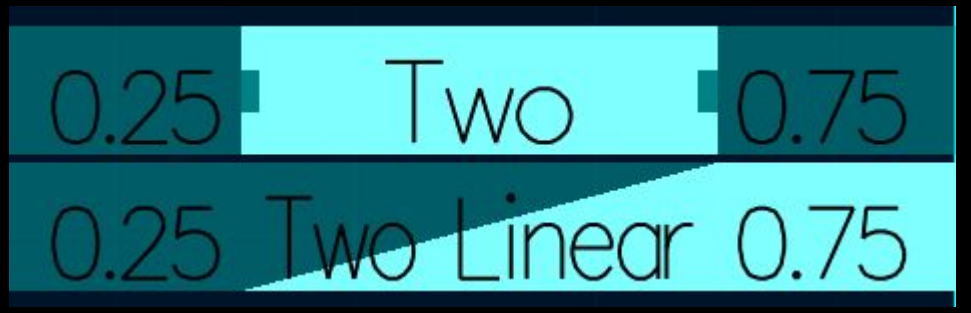




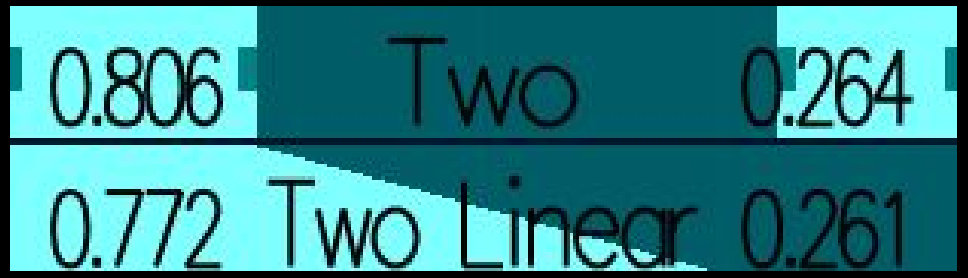




SLIDER_TWO



- Same but **2 values**
- Range / Linear
- Value change depends **where you click**
  - Closest is selected for interaction
  - Last interacted value receive keyboard input



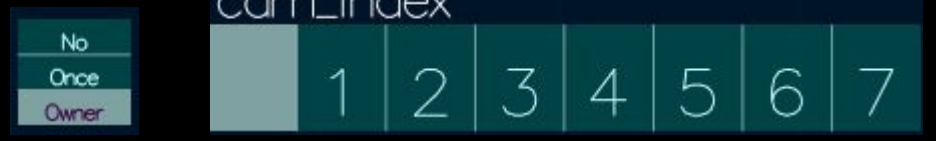
- Can be **Flipped**
  - Left value bigger than right
  - Can be seen

SELECTOR

- **Click**

  Keep mouse down

  Change with position



- Access to **StarMenu**

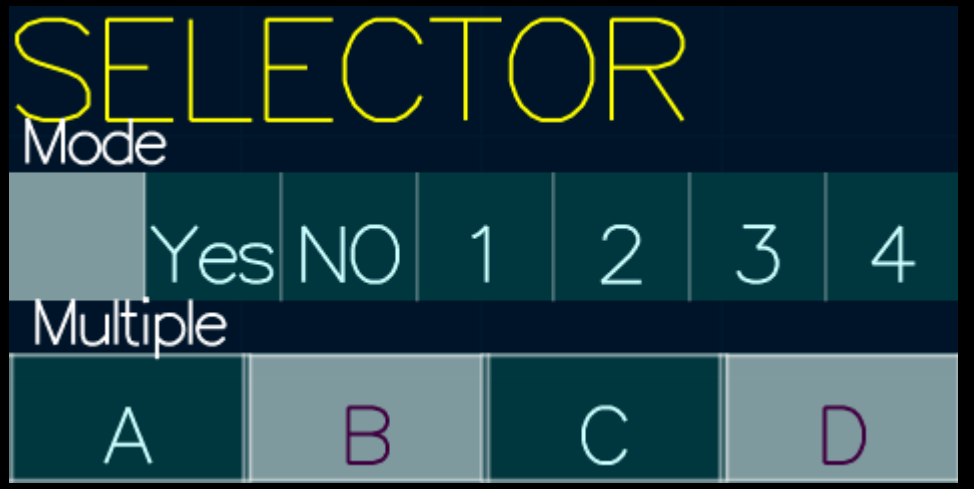
  like **BUTTON**: **Quick drag-out**

- Use **keyboard as for a SLIDER**

  + and - mainly

- **Multiple selection** an option

  display frame around each item



SELECTOR

- Slide functions for selector too

  Go up

  then angle

  Shift

  lock on values

  Ctrl or Ctrl Alt

  sensibility

- Keys as usual

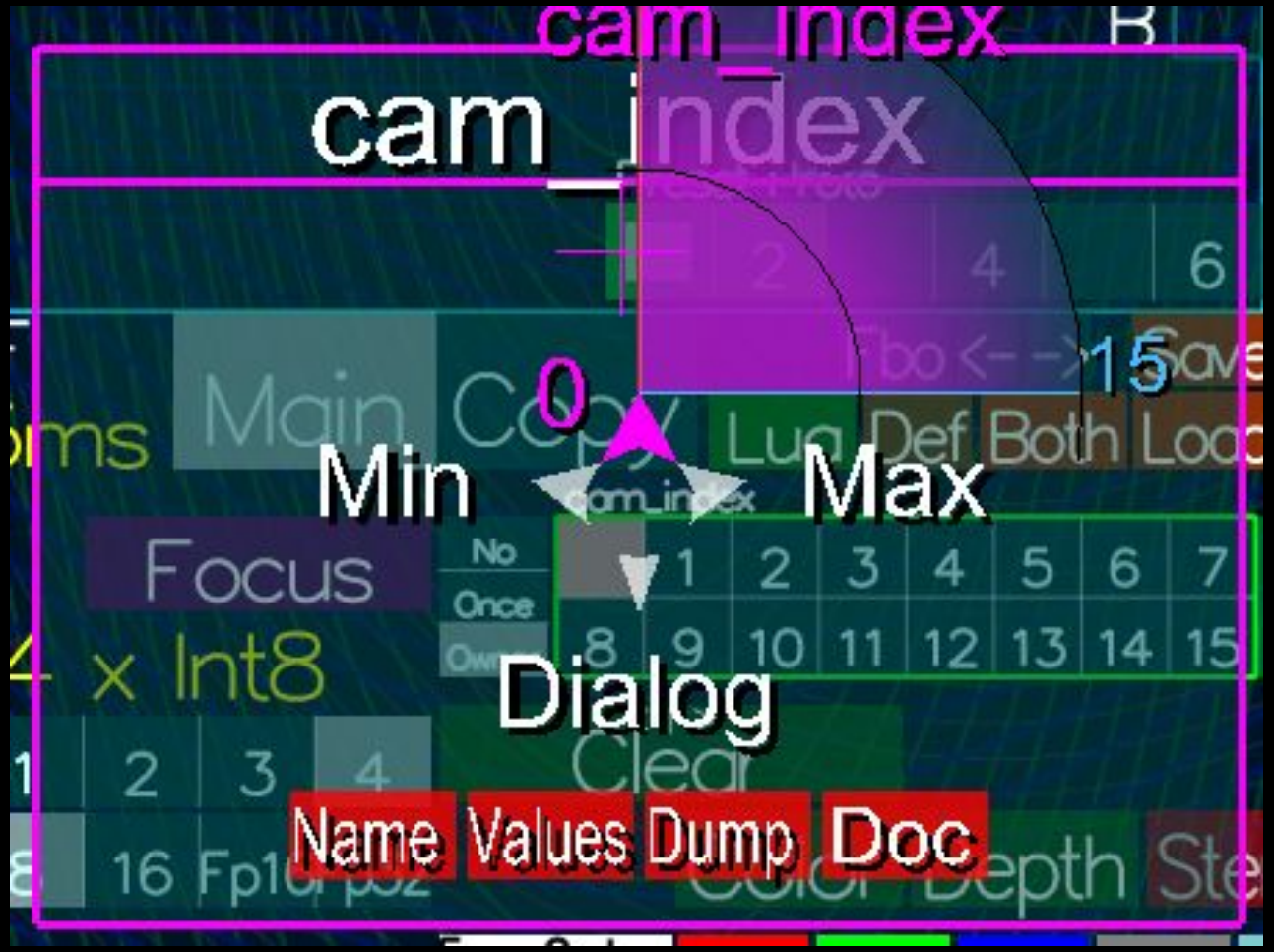
  + - * / → change

  Enter → negate

  . → floor integer

  Page Down / Up → Min / Max

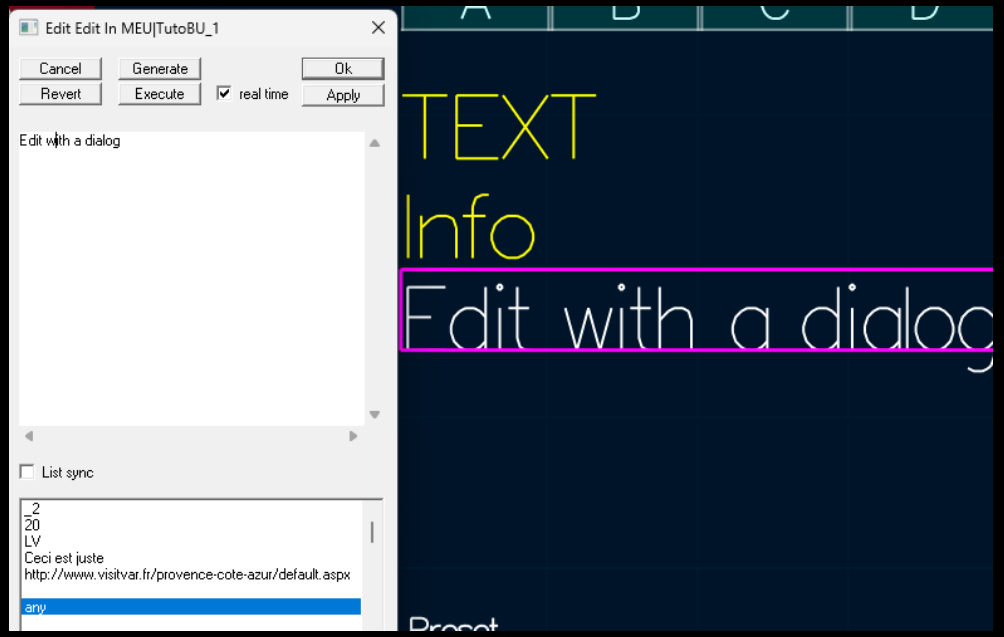
  Home / End → Default / Default inactive



BU_TEXT

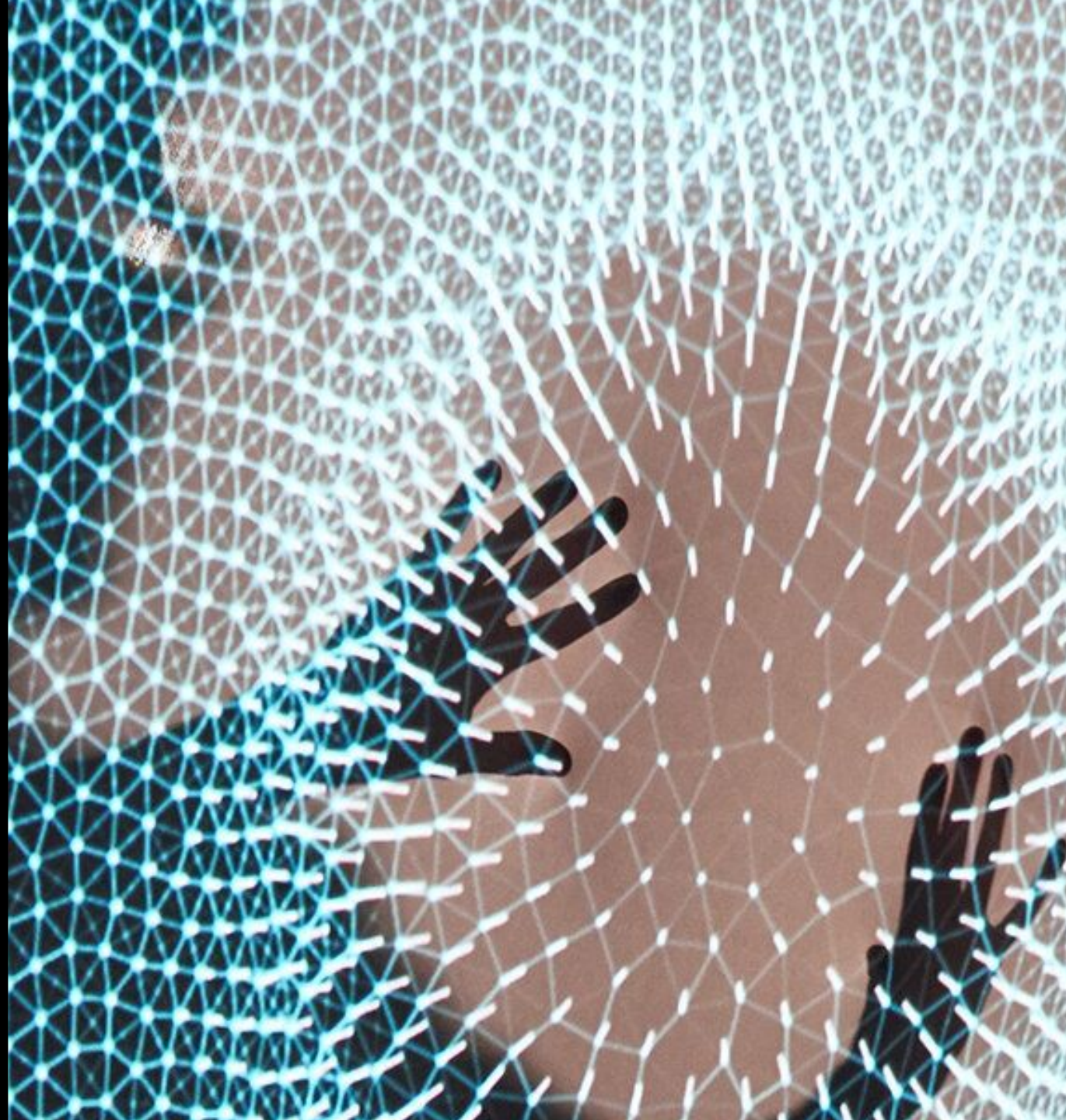
- **Editable or not**
  Double Click to Edit
- **BU_TEXT info**
  Yellow in general
  display info
- Access to Star Menu
  like **BUTTON** and **SELECTOR**
  Quick drag-out

- will Evolve
- Dialog will Evolve too
- Virtual keyboard exists but not functional
  at the moment (2025 July)



AAASeed
An introduction
Part 7:
Meet more BUs

- **Other SLIDERs**
  - SLIDER_XY
  - SLIDER_MULTI
  - SLIDER_CURVE
- **Composite Bus**
  - BUTTON multiple
  - BUTTON menu
- **Copy / Paste**
- **Undo / Redo**



Other **SLIDERs**

Select More in top Selector to see

SLIDER_XY

2 values x y

SLIDER_MULTI

Edit several x, y, size_x and size_y

Alt to Move/Resize as usual

Alt Click → **Move/Resize**

Alt Arrows → **Move**

Alt + - * / → **Size**

Double Click to toggle

Key return on current BU

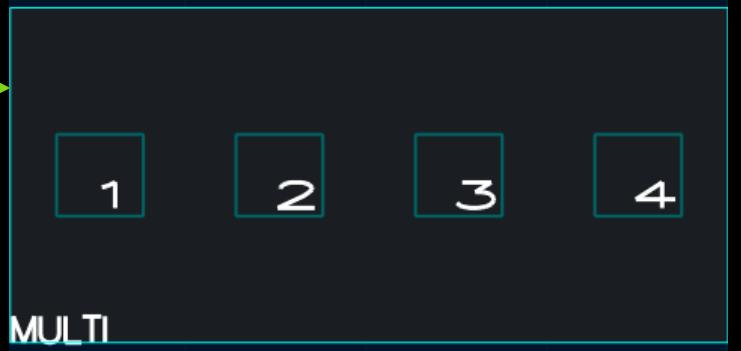
Key insert → Add point

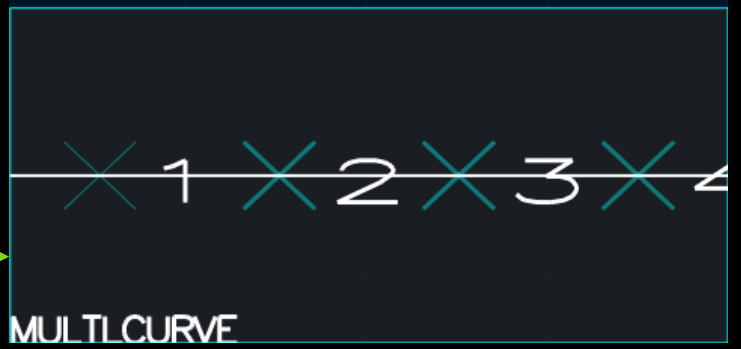
Key delete → Remove point

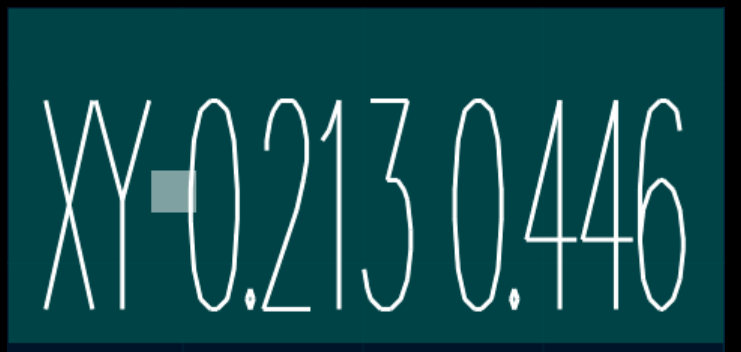
SLIDER_MULTI_CURVE

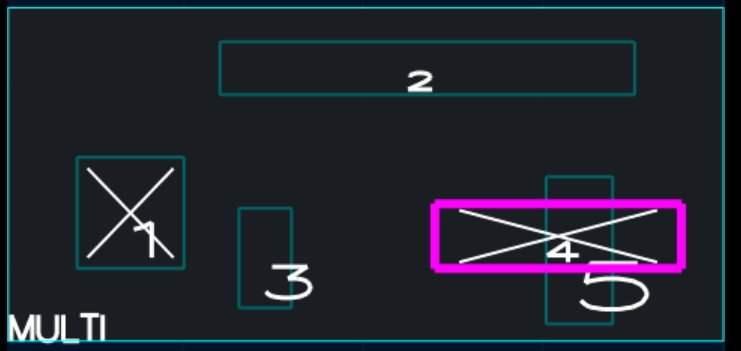
linear curve

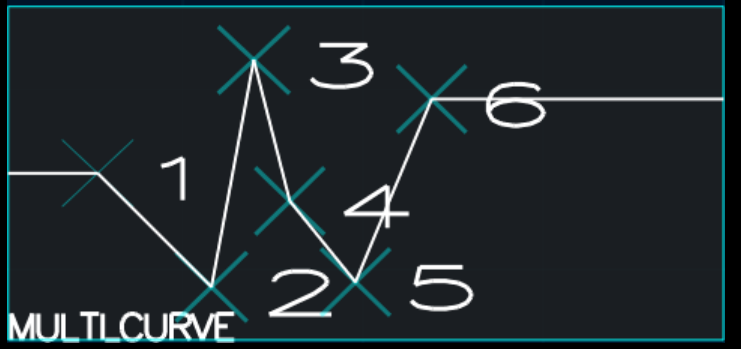
UI as SLIDER_MULTI







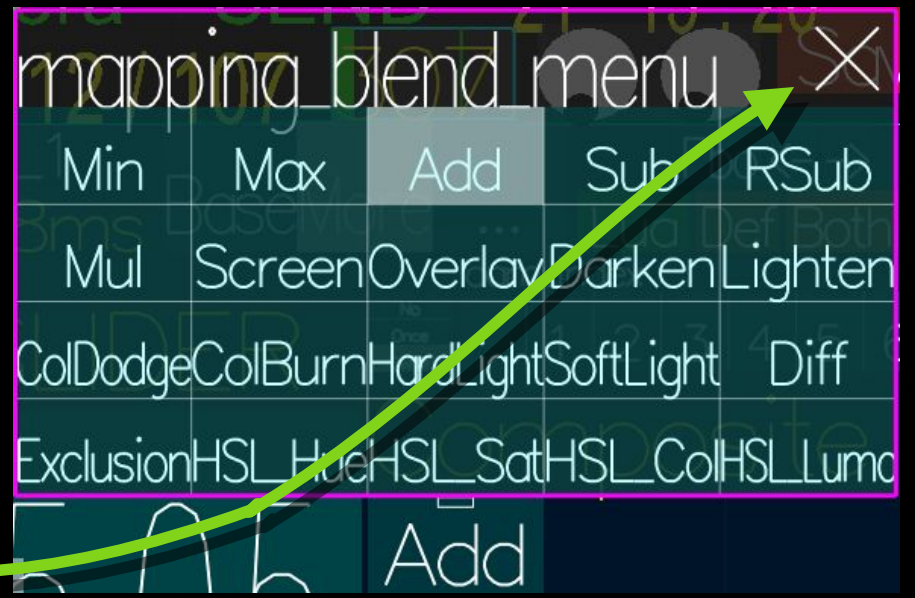




Composite BUs

- **BUTTON** multiple
  - < | > at the top
  - Use a hidden SELECTOR
  - **Click left or right to change**
  - **Same keys as selector**

- **BUTTON** menu
  - small rectangle at the top
  - Use a **SELECTOR** in an optional **window**
  - **Move window to keep it open**
    - or it disappears after few seconds
  - **Close BU** at top right
    - close the window
  - **Same keys as selector**



Copy / Paste

- Ctrl c / Ctrl v
- Acts on current BU
- Shift extend it

Copy / Paste not ONLY a single BU

but a bu_group or a MEU

see later

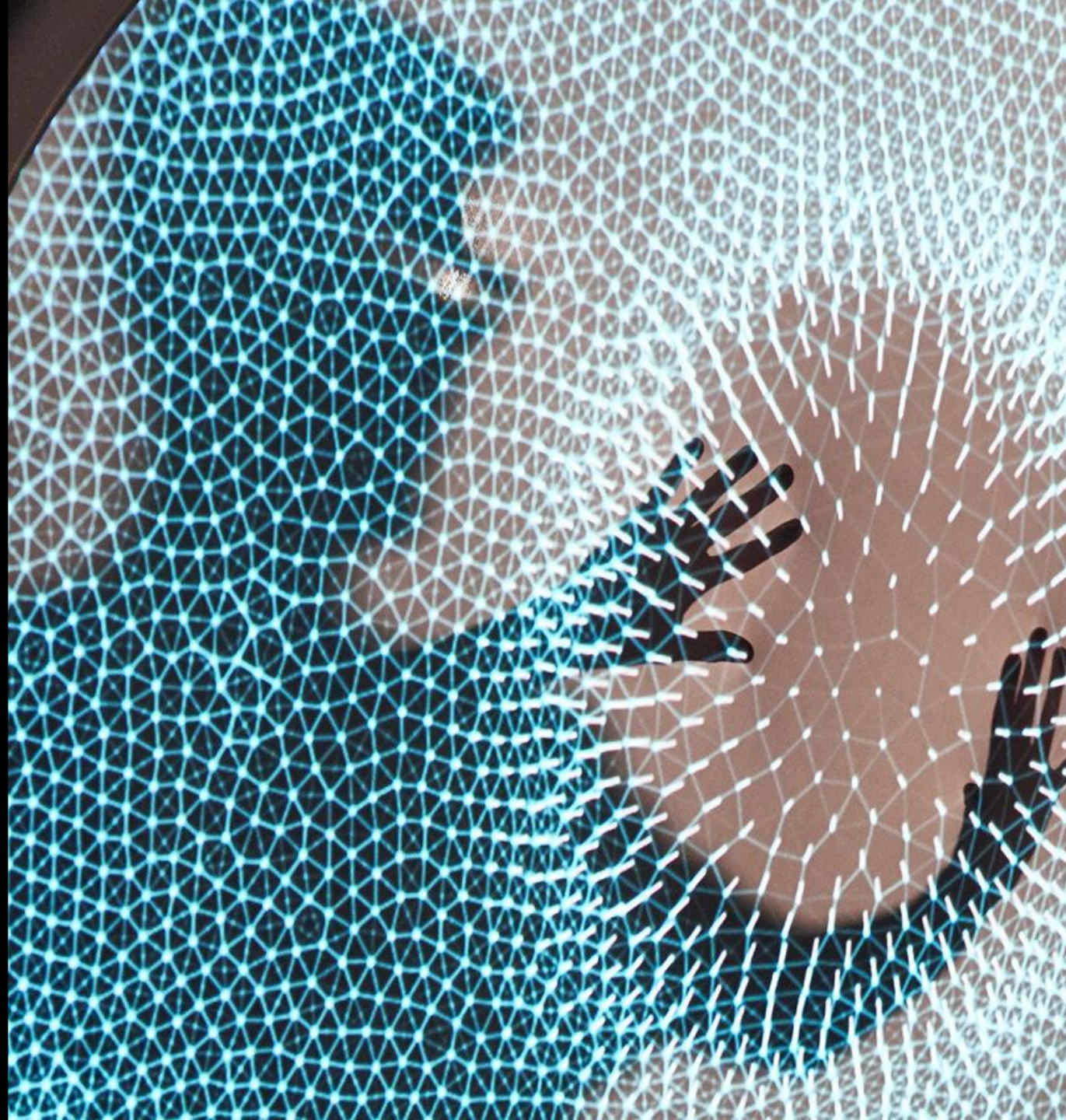
need to be refined and documented

Undo / Redo

- **Ctrl z / Ctrl y**
- **Deal with**
  - **BU values**
  - **BU size and position**
- **Unlimited**
- One more reason not to be afraid
  - try things then cancel
- **Shift Paste** Special case (2025 July)
  - treated as a **series of individual actions for now**
  - **instead of a global change**
- **Still some bugs (2025 July)**
  - **we need feedback to correct !**

AAASeed
An introduction
Part 8:
Specialized BU

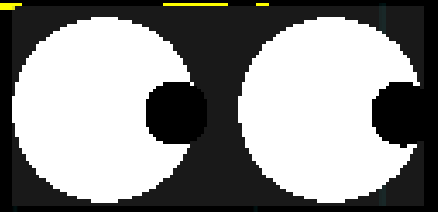
- BU_WIZs
- BU_MESS
- BU_SHOW
- Window GA
- Window BU
- Window GP



Some **BU_WIZ** (Wizard)

- **BU_ALIVE** Flip rendering (toggle with Click or F3)
- **BU_FPS** Frame Per Second FPS 153 / 145

  FPS in Flatland: can see fps with no UI
- **BU_TIME** show time 23 : 26 58
- **BU_EYE** show mouse direction



- **BU_CAM** control camera editing CAM Locked
- **BU_SEND** control synchronization between machines NO SEND
- **BU_MEM** show memory usage in MegaByte

  problem if it increase continuously 1165
- **BU_POWER** show power and plug status Power : Plugged 92%
- **BU_BLOB** show number of contacts -1-

BU_MESS

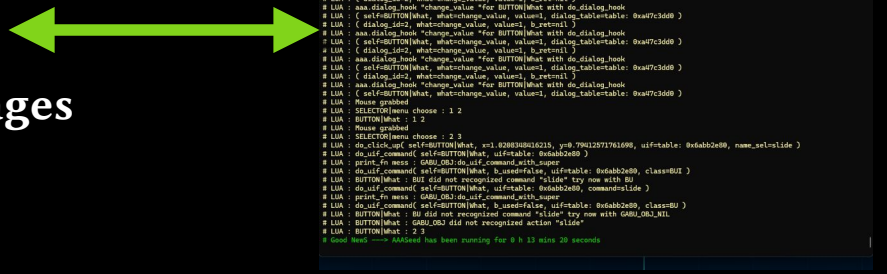
```
# LUA : BU_CAM|viz_CAM : ---------- GABU_OBJ unused key 9
# LUA : BU_CAM|viz_CAM : did not used key 9
# LUA : GARDEN|garden : try to use key 9
# LUA : GARDEN|garden : do_key( key=9 )
# LUA : do_key( self=GARDEN|garden, key=9 )
# LUA : GARDEN|garden : ---------- GABU_OBJ unused key 9
# LUA : GARDEN|garden : did not used key 9
# LUA : GA|SINGLETON : KEY NOT USED 9
# LUA : GABU.do_key_custom_def() key 9 Unused
```

Scroller

Mess

Try Double Click

Key m for **(m)essage pop up terminal window**

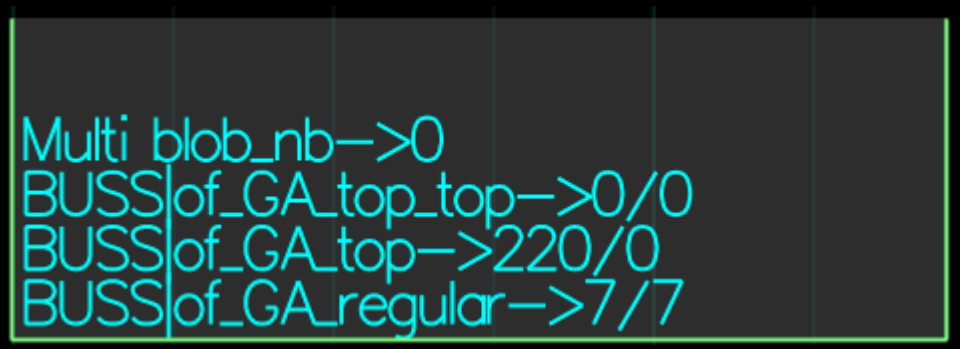


Scroller is a **Slider** to **move** in the history of **messages**

Messages color have a meaning

- **Error**
- **Trackers (devices plugged, sending information to AAASeed: Midi, Camera, Captors...)**
- **Debug**
- **Info**

BU_SHOW (Debug Tool)



More and more **Red** Message for **errors show** there

Accessed in Lua using

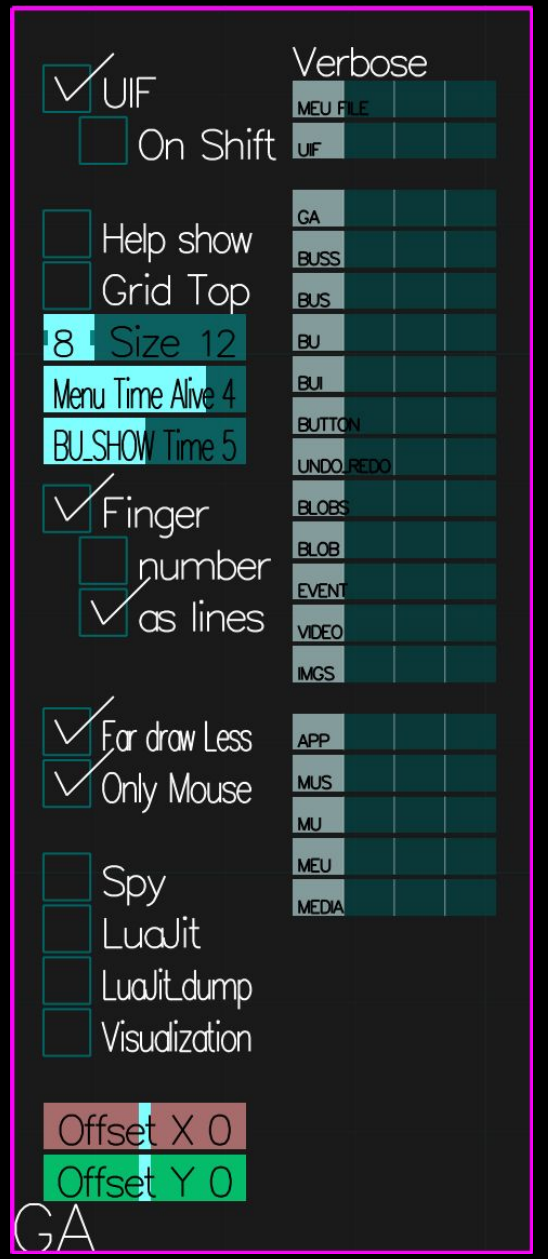
aaa.show(value, name)

self:show() for

Window GA first look

- **G**lobal **A**ction settings
- For now
  - **UIF** (**UI** Fast for StarMenu) → enable the **StarMenu**
    On Shift → Enables StarMenu on Shift Click Only
  - **Help Show → same as F1 / Ctrl h**
  - **Grid Top → draw a 8x8 unit grid** on top of the UI
  - **Size → set the 2 sizes used by Ctrl Tab**
  - **Menu Time Alive**
    how long a button menu stay on
  - **BU_SHOW Time**
    how long an element stay in BU_SHOW
  - **Finger**
    What I use in the tutorial videos to **show the mouse**
  - **Far draw Less → simplify** drawing with mouse distance
    a way to optimize UI display
  - **Only Mouse** → discard multitouch input
- **Rest** will be **detailed later,** mainly used by **developers**
  Avoid if you don't know what you are doing

- **Ctrl S, Esc/Esc Save it**
  file AAA_Ga.bus in the APP folder



Window BU

- **B**ox **U**ser settings
- Most of these Options for developer

  graphic debug

  will have a specific tutorial
- Line Width → Text line width

- **Ctrl S, Esc/Esc Save it**

  file AAA_Bu.bus in the APP folder

Window GP

- **GP** come from **G**arden **P**arty
  - **first time the MEU/MU interface was used**, it was during an **Utram** (french company) **G**arden **P**arty.
- Mainly drawing switches

  use for debug or optimization
- Render Chain

  Toggle the display of the render chain (See Part 10 Render Chain
- Grid Alpha

  transparency of the grid displayed under the UI

  drawn only if superior to 0
- MEU Timings

  Toggle MEU timing measurements

  if an application render a lot of MEUs it could alter the frame rate, and so, setting it to off will make the rendering faster.

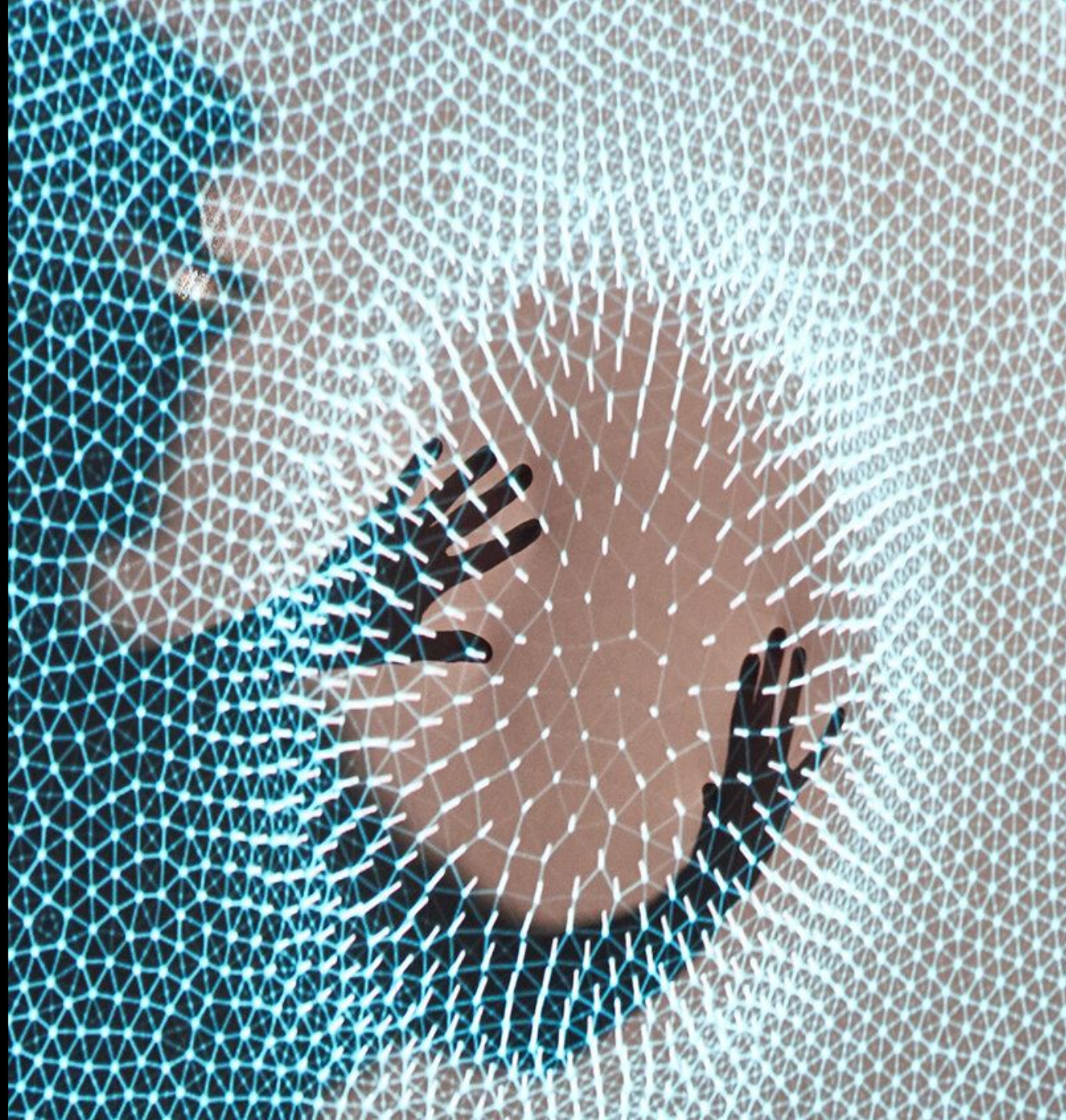
  Also when an application is stable and in exploitation, the MEU timing measurements are just a waste.

- Ctrl S, Esc/Esc Save it

  file AAA_Gp.bus in the APP folder

AAASeed
An introduction
Part 9:
MEU

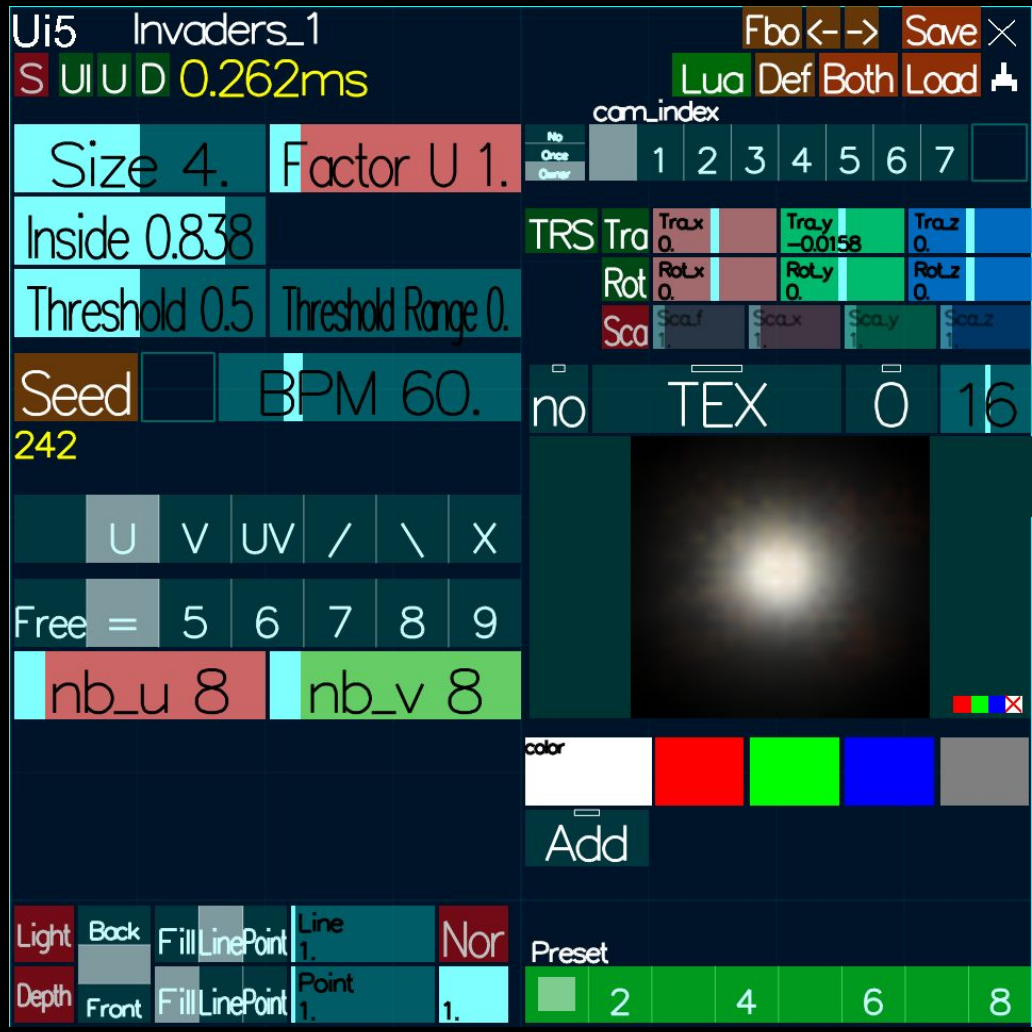
- MEU / MU
- MEU and Uix
- More on MEU
- MEU Bar
- MEU Preset
- MEU StarMenu



MEU Module **E**ditable **U**nit
MU Module **U**nit

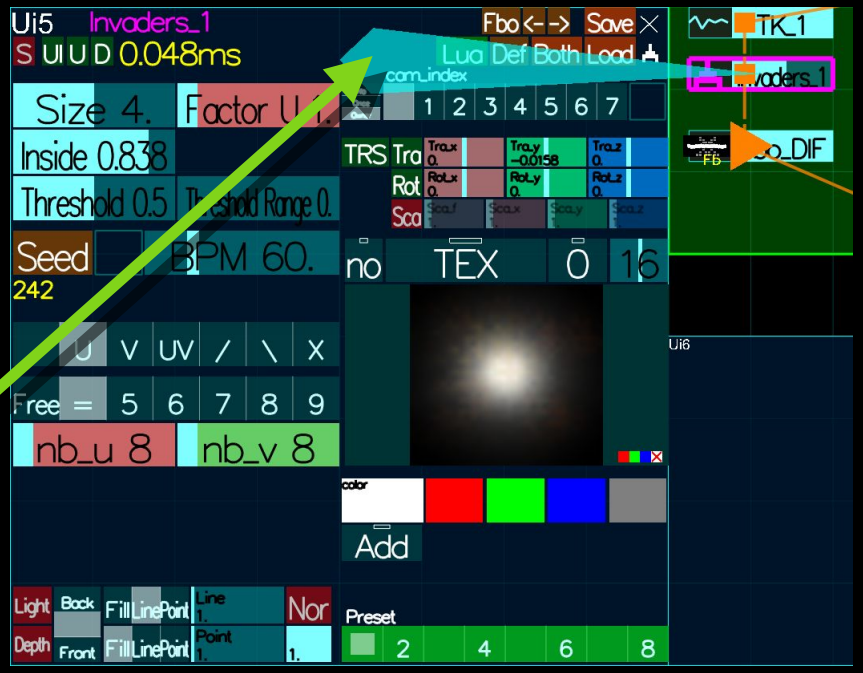
- **MEU** is the base to encapsulate and manipulate functionality
- **MU** compact part of the **MEU**

- **Click** on **icon**
  - **Drag to move**
  - **MU** become **current BU**
- **Click** on **slider**
  - **StarMenu**
  - **slider** become **current BU**
  - **slider value** is the **alpha of the MEU/MU**
    - Slider value **more than 0 ⟶ On**
    - value control **transparency** when applicable
  - double click ⟶ slider edit dialog
- **State and Position** of the **MUs** control the rendering process
  - see next chapter
  - vertical position first then horizontal position:
    - **bottom to top then left to right**



MEU and Uix

- Click on **MU icon** → **MEU** in **Ui5 / Ui6**
  - **Ui5** is the default
  - **Ui6** is the default for **Folder MEUs**
- **Link shows corresponding MEU / MU**
- **MU StarMenu** to choose Uix
- **MEU StarMenu**
  - **click on backgound of Uix**
  - **nearly the same as MU StarMenu**
- Close **BU** at top right
  - detach MEU from its Uix window
- Red Background ↔ not rendered



- Uix are technically **BU_MEU**
- Drawing change with distance
  - this is a speed optimization

More on MEU

- **MEU** is a **module** of functionality,

  in computer terms it is an **object**

  it **usually renders** but can be otherwise:

  receives/send data

  analyses image

  control a device (e.g. plotter, projector, Dmx, Arduino...)

  ...

- **2 main methods** (function) are called every frame by a **render() method**
  - `update()` prepare so draw() will be as fast as possible
  - `Draw()`
- **1 method** is called when the **MEU Ui is visible**
  - `update_ui()`

MEU real nature, a peek for now

- It is a **Lua object**
  - Name is MeuType_InstanceName
  - **a lua Script for each Type** (people say also class or prototype)

- It can uses **C++ AAASeed Objects** (c_obj_ui)
  - c_layers or c_module most of the time
  - Accessible via its flatland parameterw and some dedicated lua methods
  - **Focus** Button in StarMenu

- It lives in a **directory**
  - Everything is readable and can be edited
    - less and less needed
    - but some like it
  - **Dir** Button in StarMenu

MEU Bar

- TutoBU_1 **Title** (white or Violet if MU is current BU)
  Meu's name made of **type** (left) and **instance** (right):
  TutoBU_1 means **MEU type TutoBU instance 1**
- S UI U D **buttons**
  switches to activate/deactivate
  Send update_**UI** Update and **D**raw
- 0.032ms **Execution time** in millisecond
- BaseMore ... is a SELECTOR named TAB
  switch between different block of MEU **Ui**
- **Close** button **detach** the **MEU** from its Uix

MEU Bar

- Lua **Open** the **lua** script that the **meu** use in an associated text editor
- Load **Load** only this **meu** state
- Save **Save** only this **meu** state
- Def Trigger a **Definition**, in fact a redefinition of the **MEU** (its UI in particular)
- Both Trigger a **Definition** and a **Load**

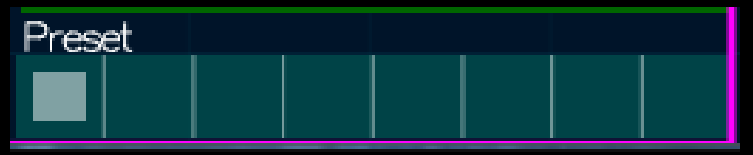
- Fbo Open the used **Fbo** (more soon on Fbo)
  in the previous **UIx**
- <- -> lets you navigate in the rendering chain (more soon)

MEU Preset

- At the bottom right of the **meu**
- **Load/Save the state of a meu**

  in fact load/save the values of meu's **BU**s

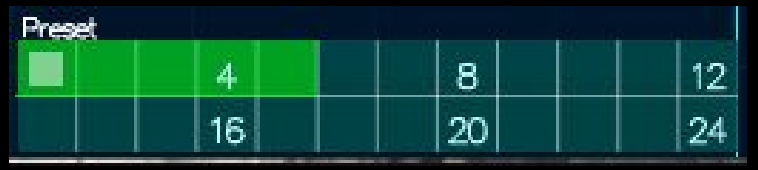
  not always all, at the author discretion



- **preset defined ↔ green color**

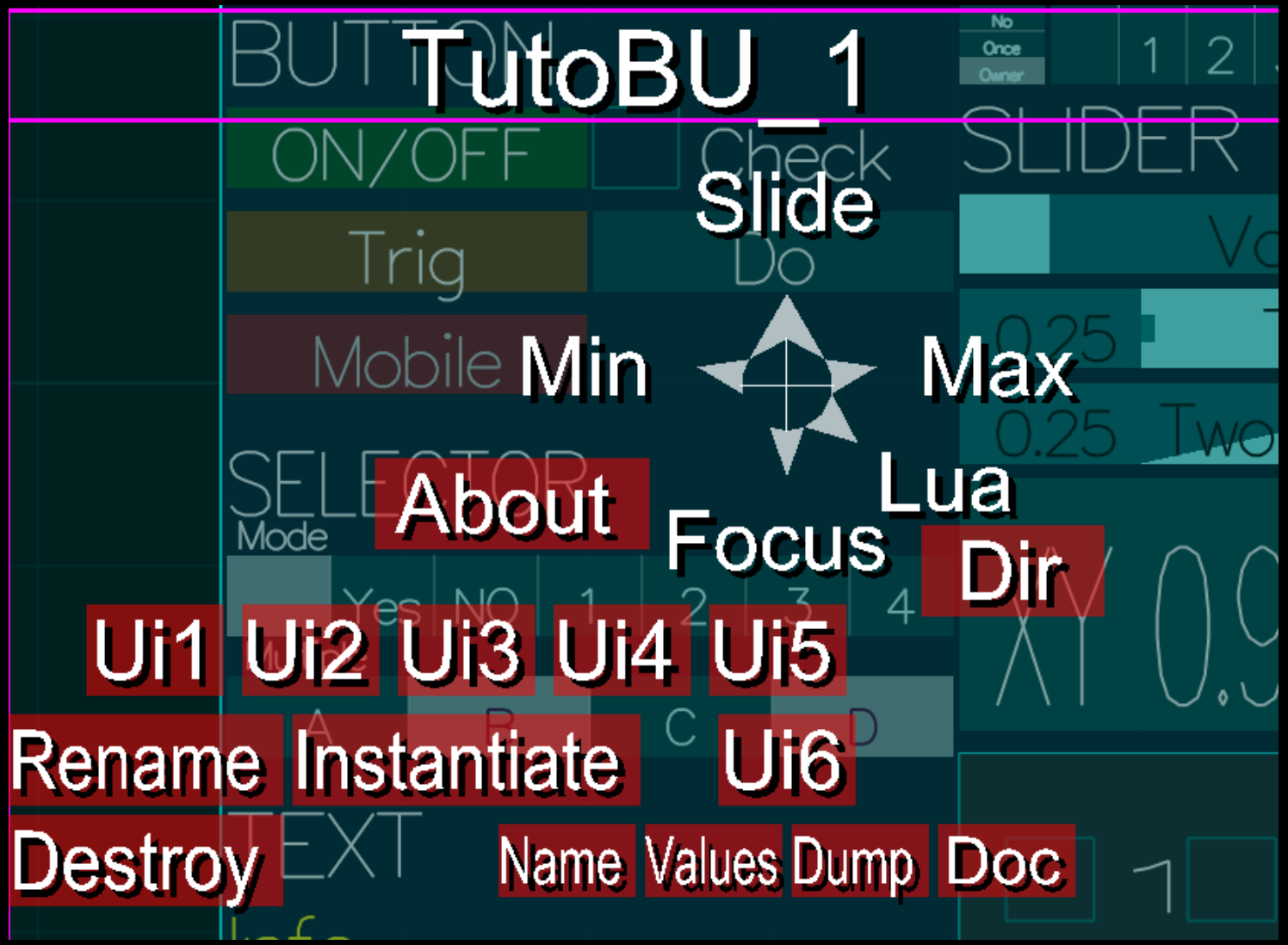
- **Click** **Load**
- **Ctrl Click** **Save**
- **Ctrl Alt Click** **Delete**
- Fixed number by MEU Type

  defined in the code for now (2025 July)



Developer Note: here the code is meu:get_preset_nb() return 24 end

MEU StarMenu



- **Click MEU background or MU slider**

  Slide

  Adjust the alpha

  control the **MU Slider**

  Focus

  Access to **Flatland**

  C++ part of the MEU

  Lua

  Access to the **Lua script**

  Using **default editor**

  Dir

  Opens the **Instance folder**

  Uix

  Choose which **Uix window get the MEU**

- **Described in Part 11**

  Rename

  Type can't be changed

  Destroy

  with Confirmation Dialog

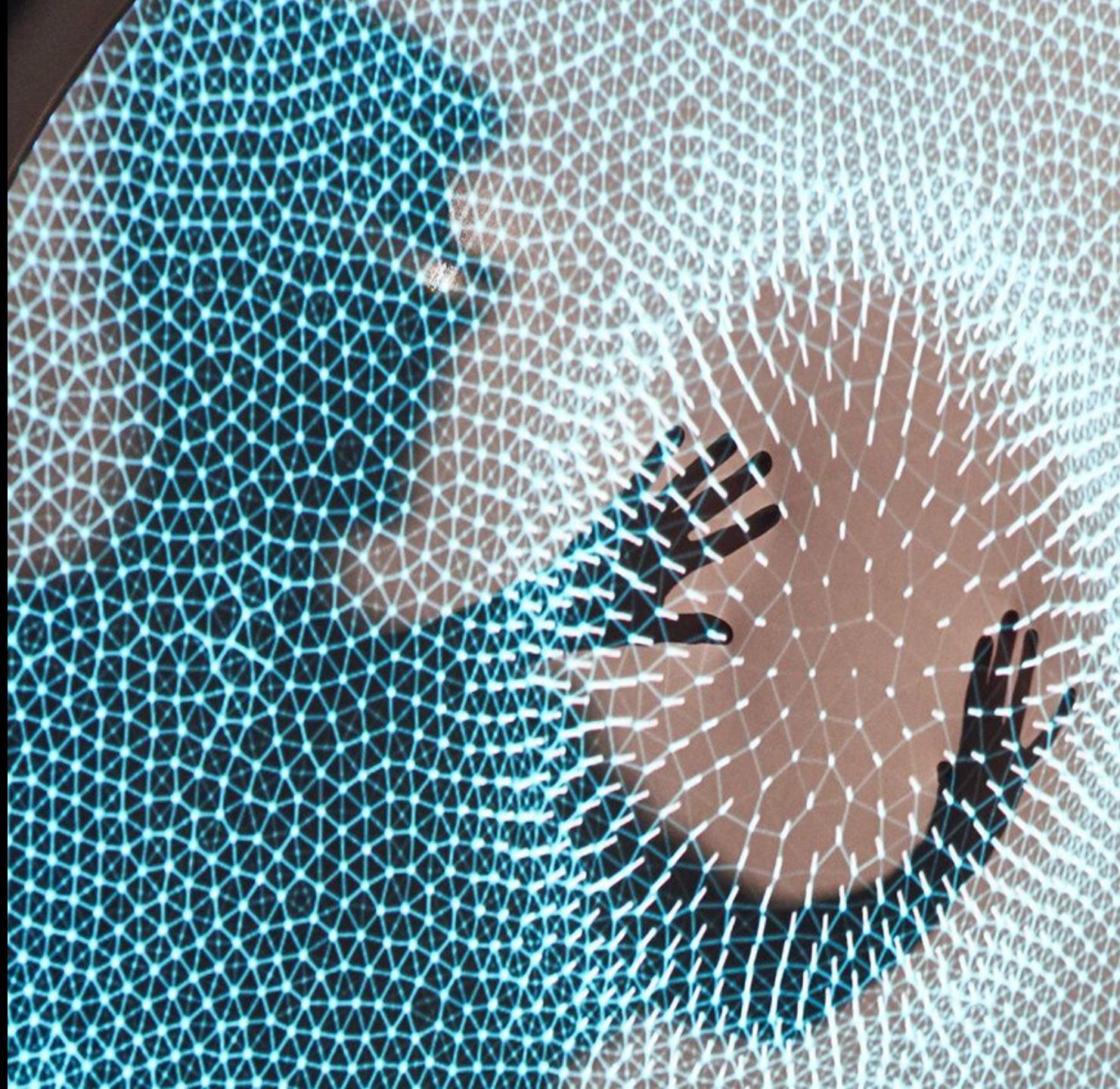
  Instantiate

  Duplicate

  Type Name can't be changed

AAASeed
An introduction
Part 10:
Render chain

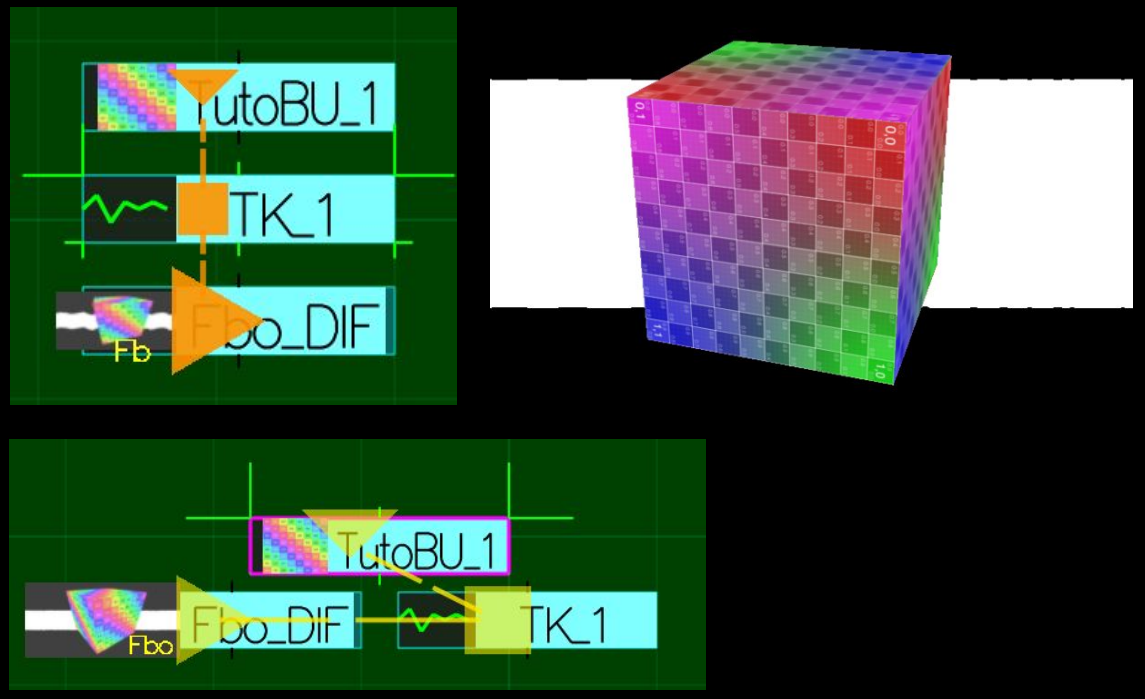
- Rendering chain
  - **Order**
  - **Alpha / Opacity**
  - **Visualization**
- BU_RECT
  - **Order**
  - **Value**
  - **StarMenu**
  - **Move**



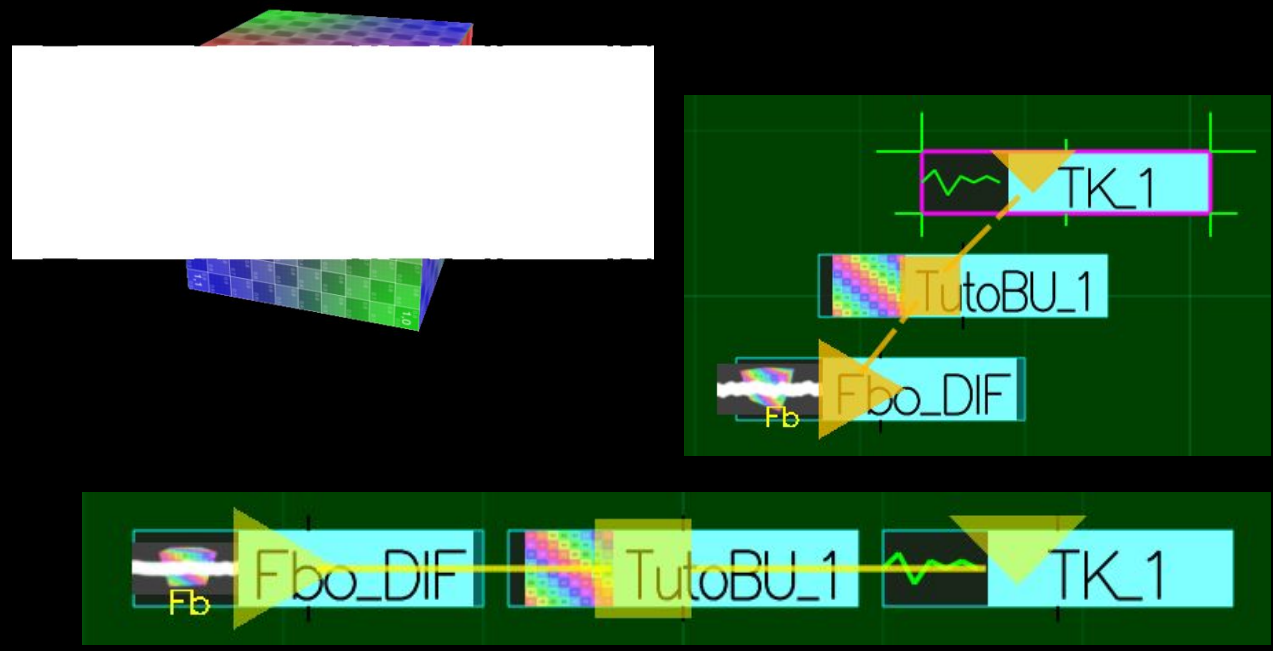
Render chain: Order

- **Position** of the **mu** control the **execution/rendering order**
- **bottom to top**, then **left to right**

TK_1 rendered first, then TutoBU_1



TutoBU_1 rendererd first, then TK_1



Render chain: Alpha/Opacity

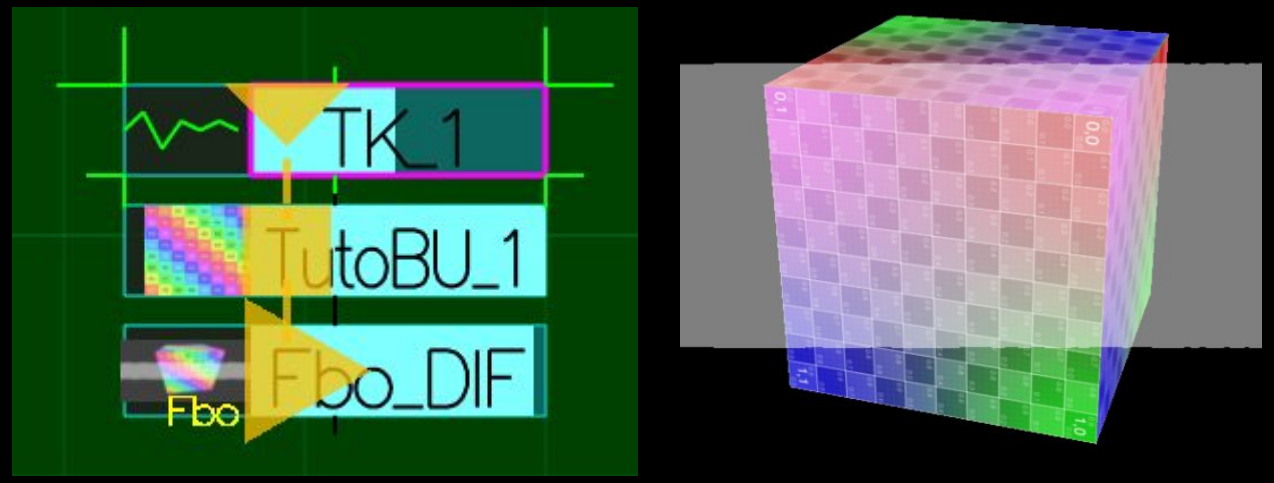
- **mu slider value is the Alpha of the meu's rendering**

  1 → **opaque**

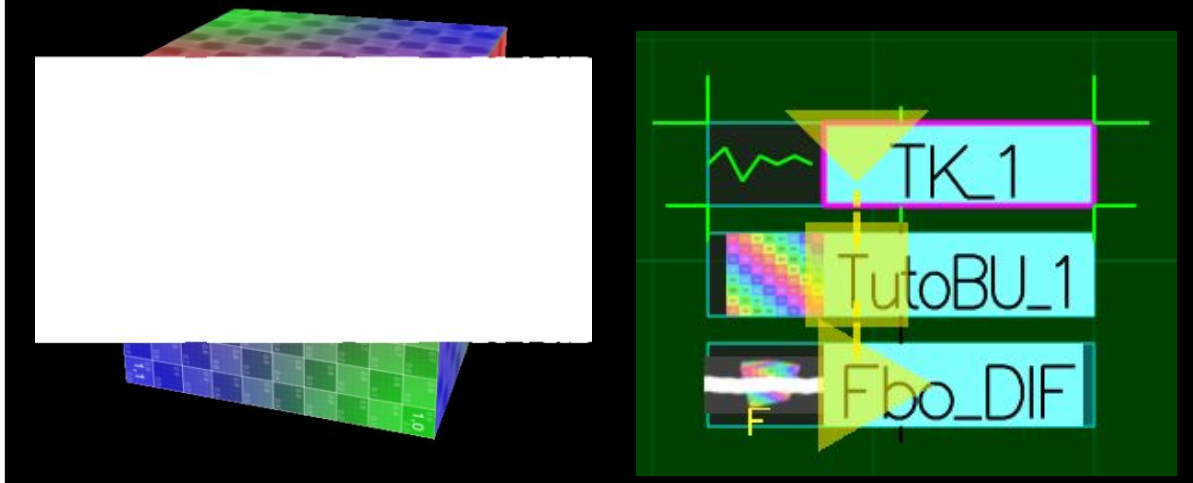
  0.75 → **25% transparent**

  0 → **100% transparent** → **meu Off**

TK_1 is 50% opaque

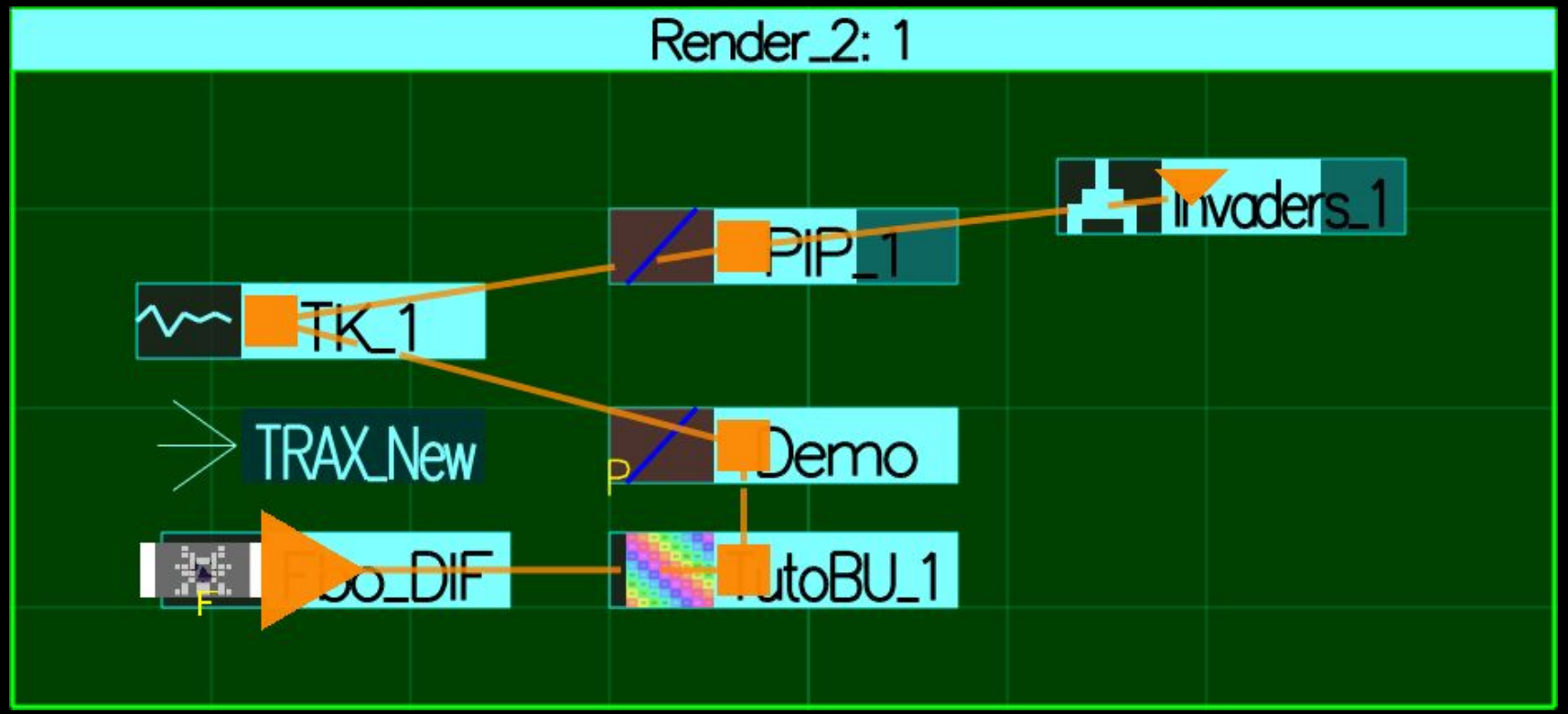


TK_1 is 100% opaque



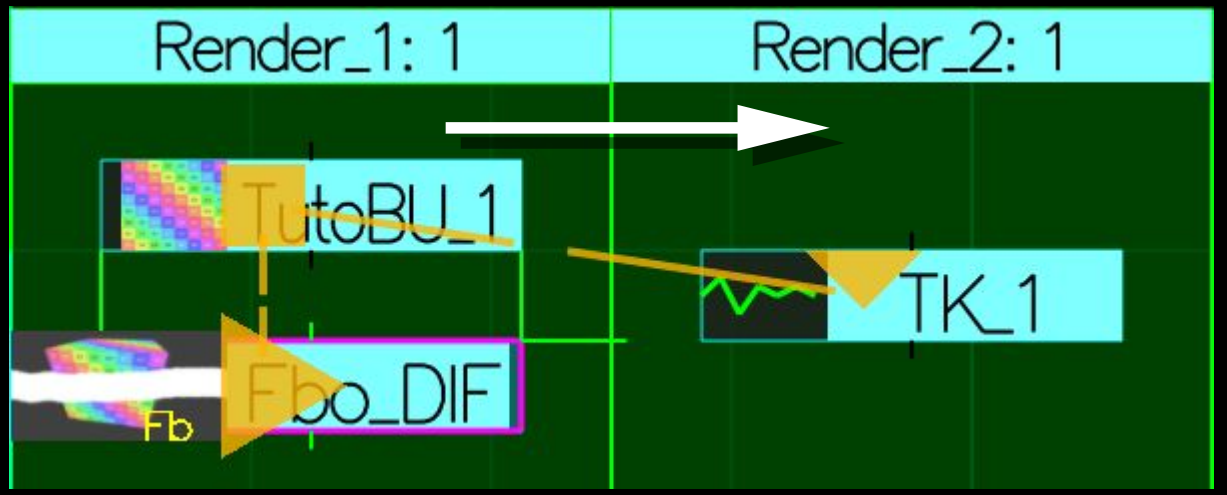
Render chain: Visualization

- Orange line shows the render chain (the execution order)
- **Just a visualization not a cable !**

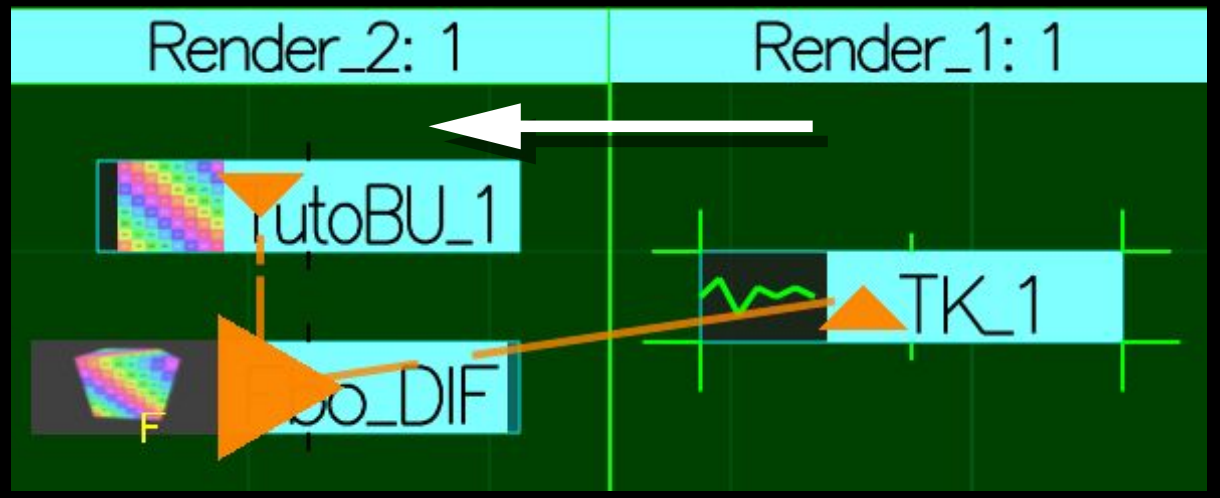


BU_RECT Order

- First: Order of the BU_RECTs
  - the number in the name
  - to change it
    - StarMenu
    - Keyboard
      - Ctrl Up / Down



- Second: inside BU_RECT
  - Bottom to top
  - then
  - left to right



BU_RECT Value

- Control all mu inside:

  it multiply their Alpha.

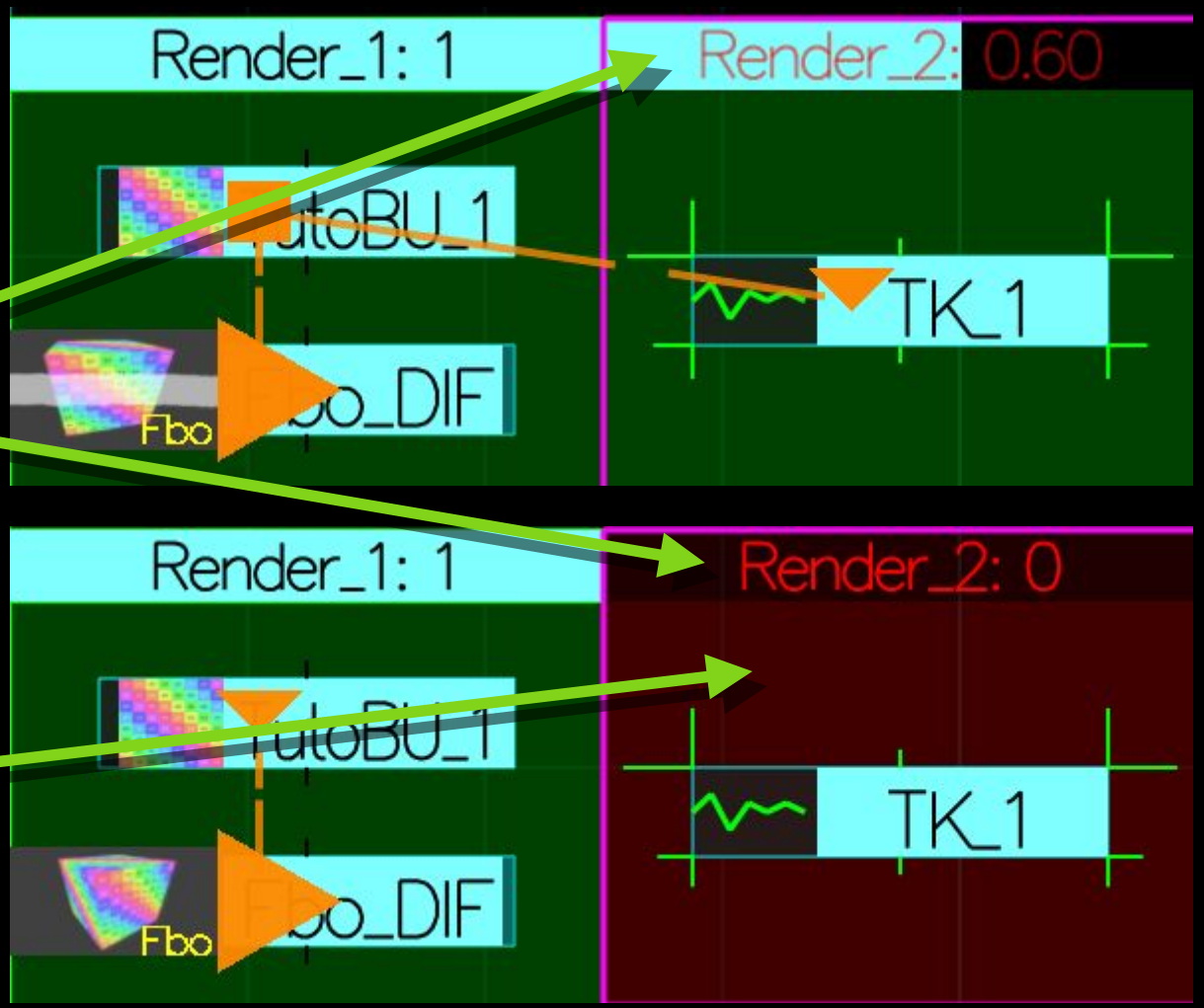
  Act as a group Alpha.

- different from 1 red text

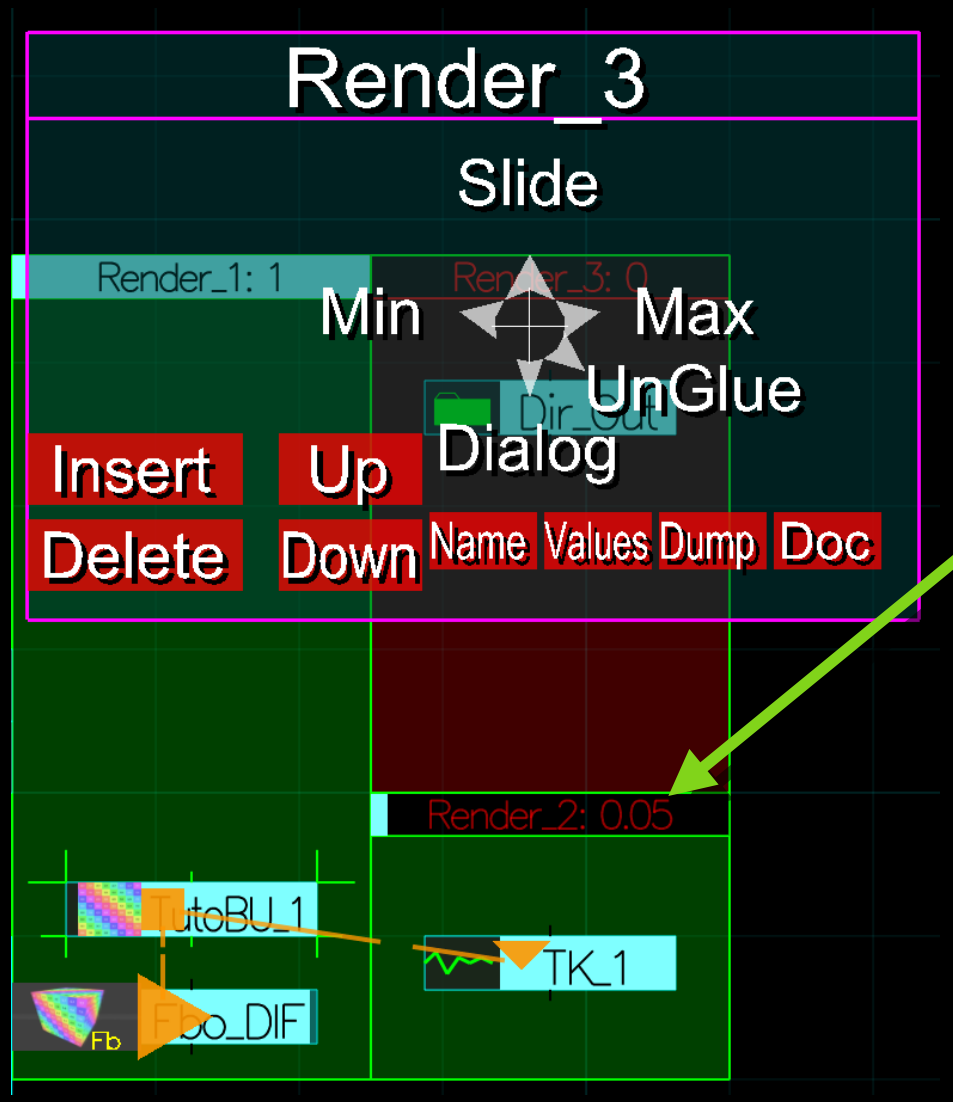
- inferior or less then 0

  no mu rendered

  red background



BU_RECT StarMenu



- **Slide** to change value

  value is Alpha for BU_**RECT**

  shown like a Slider at the BU_**RECT** top

  Red when not 1

- **when current BU, receive keys**

  same keys as a slider

  Ctrl Insert / Delete → Insert / Delete

  Ctrl Arrow Up / Do → Up / Down Order

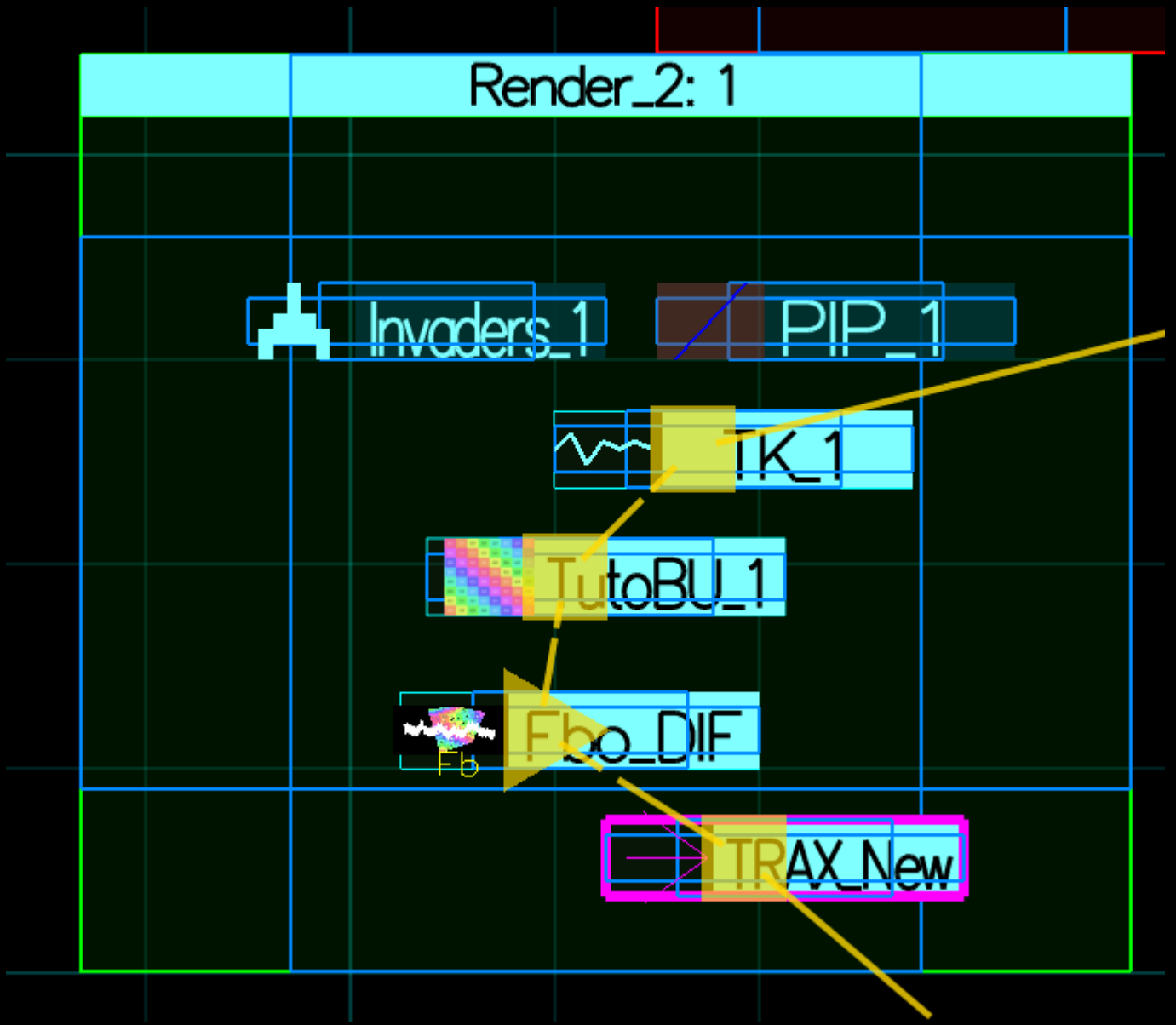
- **Glue / UnGlue**

  Affect MUs inside when moved (next slide)

BU_RECT Move Resize

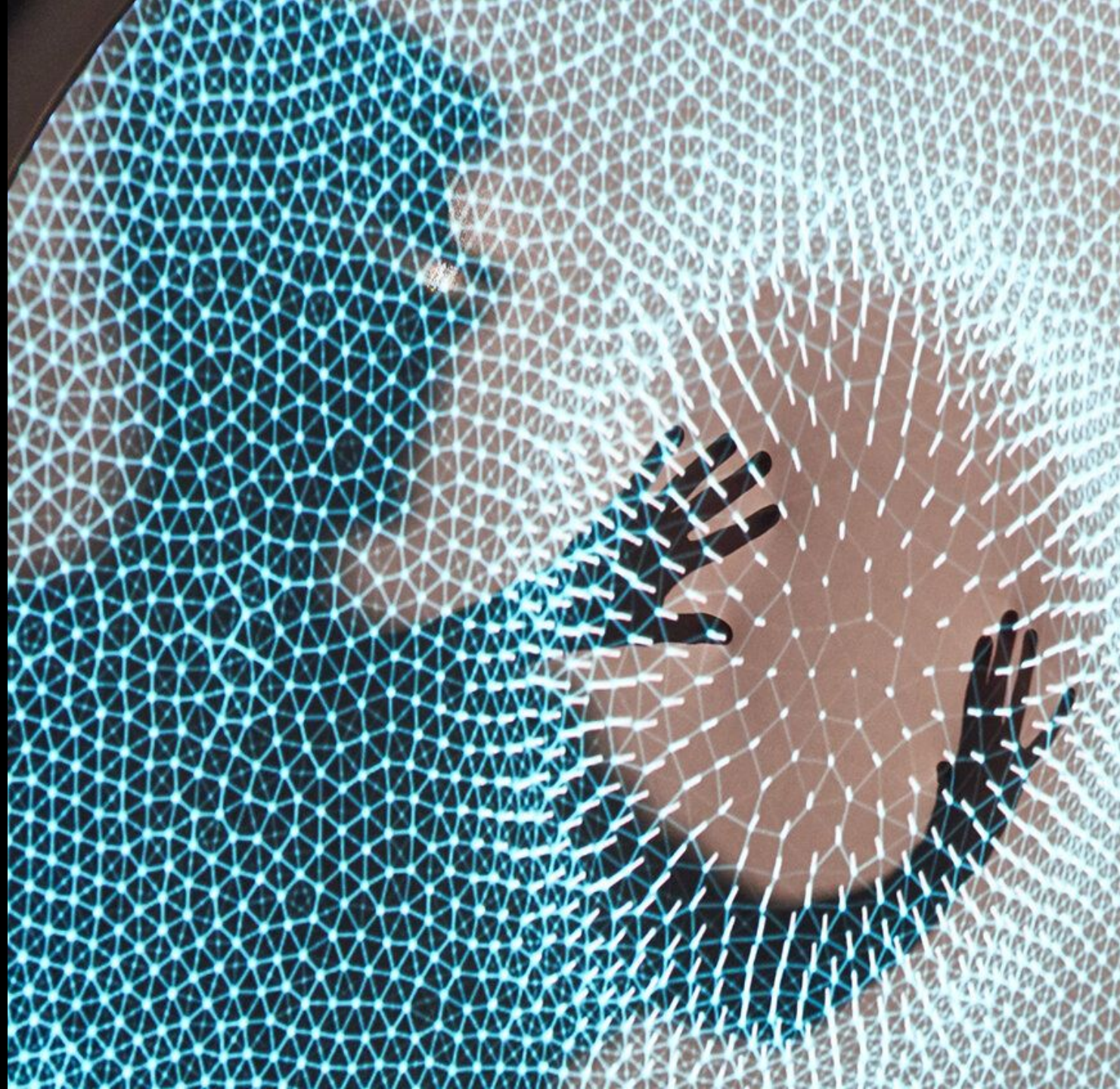
- Move Resize as always
  - Hold Alt
    - Drag in Zone
    - Keys *+ / - and Arrows

- Mu inside BU_RECT
  - center of Mu inside
  - On Move
    - Glue mode
      - Mu move
    - UnGlue mode
      - Mu stay in place
  - On Resize
    - Mu stay in place



AAASeed
An introduction
Part 11:
MEU manipulation

- Seeing MU
- Finding MU
- Moving MU
- MEU_DIR
- MEU name
- MEU Instance operation
  - **Rename**
  - **Destroy**
  - **Instantiate**
- Meu Save
  Save All
- Create New APP



Seeing MU

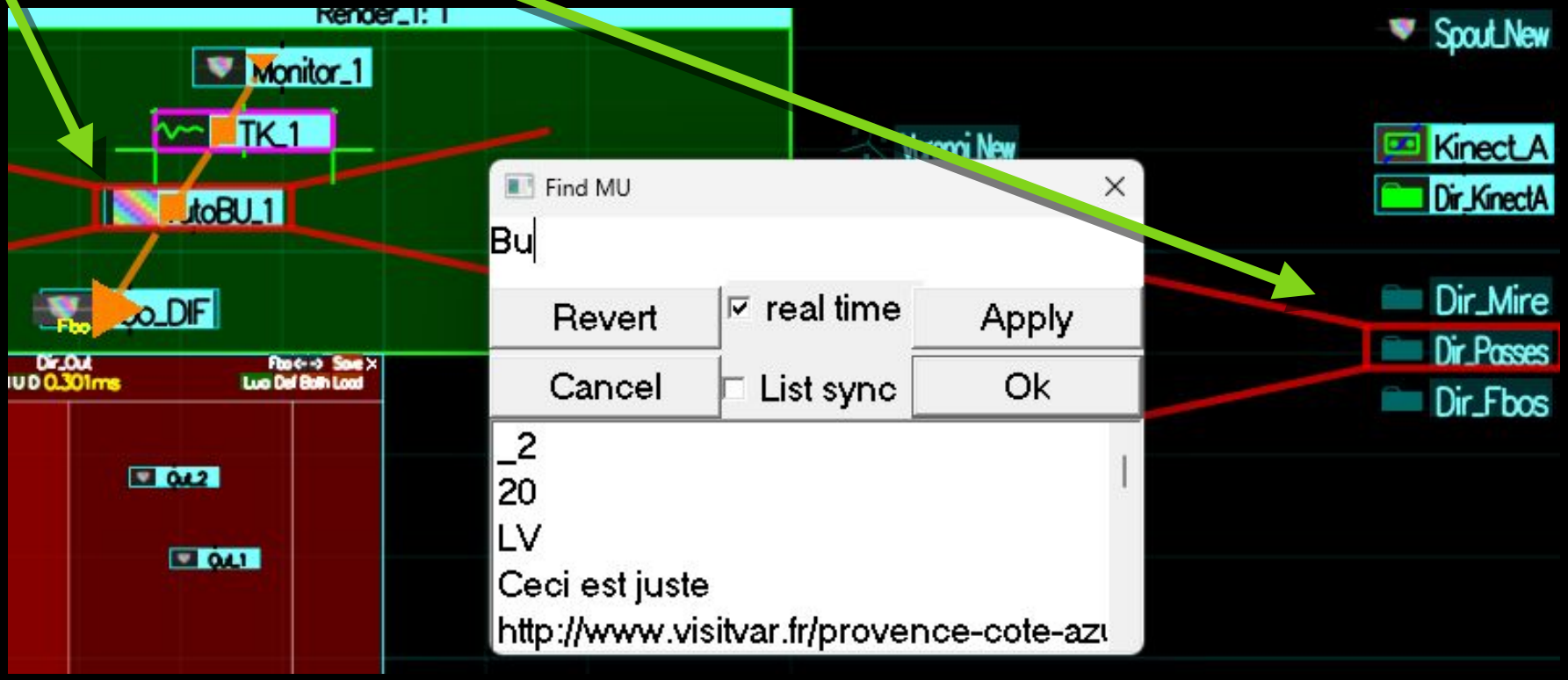
- **3 Buttons to select** what **MU are drawn**
  - Simplify the display while keeping Mu around
  - These **buttons** are **movable**
  - **Unused Button Position is special**
    - **New meu appears under (more later)**

- **Hidden** button **control display** of **MU with hide attribute**
  - **to change MU hide attribute**
    - **MU StarMenu Hide/Unhide Button**
    - **Shift Ctrl** on **Icon**
  - **Hide On → red background on MU**

- **Unused** button **control display** of **MU** unused (not rendered)
- **Used** button **control display** of **MU** used (rendered)

Finding MU

- **Ctrl F → (F)ind MU dialog**
- **highlights matching MU** (case insensitive inclusion of text in MU)
  or folder containing matching MU



Moving MU (reminder)

- Move Resize as always
  - Hold Alt
    - Drag in Zone
    - Keys *+ / - and Arrows

- Move by dragging the MU Icon

- Move by moving containing BU_RECT in Glue mode

MEU_DIR

- **Encapsulates other MEUs**
  - **regroup**
  - **simplify the display,**
    - **w**hile keeping MU around
- **It have a local Render chain**
  - **same order as for BU_RECT:**
  - **Bottom to top then left to right**
- **Open in Ui6** by default
  - More **easy to navigate the MU inside**
- **No BU_RECT inside**
  - **a central green area**
  - **2 optional areas A and B with a switch**
  - **rest is storage for MU**

- **Drag MU out → move MU Out**
- **Drag MU in → Move MU In**

- MEU_DIR use a local **AAA_MEU folder**
  - **to store its MEUs**

MEU name

- **Structure** of **MEU name** is

  MeuType_InstanceName

  first '_' separate Type and **Instance**

  we can say Type_Instance for simplicity
- **Each Type** (we say **class** or **prototype** too) have its **own Lua Script** and so its **own interface** and **behavior**.
- **Each Instance of a Type is a lua object** (it have it's own data) **use the sane Type lua script.**
- **Only letters, Numbers and underscore**
- The Meu have a corresponding folder in the file system using the same name Type_Instance.

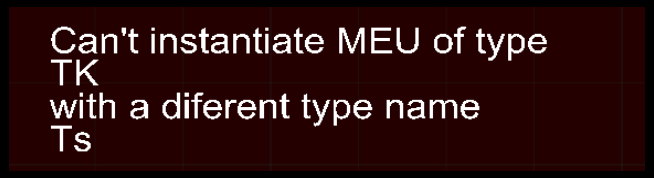
  So **Meu name have to be unique in a same directory**

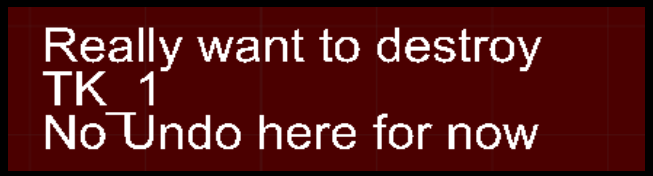
  StarMenu "Dir" to open it in the file explorer

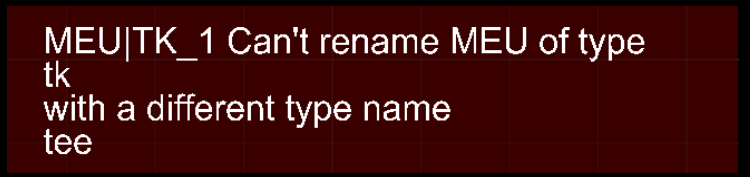
  you will find the preset file there for example

MEU Instance operation

- **StarMenu**
  - **Rename**
    - **change the name of the Instance**
    - Type part can't be changed
  - **Destroy**
    - **delete permanentlt the instance**
    - **with Confirmation Dialog**
  - **Instantiate**
    - **Duplicate, create a copy with another name**
    - Type Name can't be changed







- Drag by the icon and hold Ctrl when releasing the mouse Button: Instantiate
  - In the same folder
    - an instance is created adding the number of instances of this MEU type to the name
  - In a different folder
    - a new instance is created with the same name that the original

MEU Save

- **Button Save** at the **top right of MEU bar**
  - **Save this Instance of the MEU**
    - **the C objects (c_obj_ui) associated with the MEU**
      - **in file fx.aaa_layers_all most of the time**
    - **The state of the BU making the interface of the MEU**
      - **in file preset_0.plua**
      - **other preset are saved or deleted when action are done on preset**
    - **Whatever associated data the C object or the Lua deal with**
- **If MEU is a MEU_DIR all instance inside are saved too**
- **It is an unusual strategy but it proves very useful**
  - **you can experiment and save only what you want**
  - **eventually use Load on some MEU to reset them to their save state**
- **Button Save All**
  - **Can be anywhere this is a mobile Button**
  - **Same as doing a Save on every MEU instance in the App**
  - **Useful when you want to make sure to save everything**

- **Do a Ctrl S or a Menu/Save Globals or Quit with Double escape**
  - **to save the rest**
    - **the global level everything not in a MEU**

Create New APP: Duplicate folder

- **APP is stored in a folder**
- **Example: duplicate APP_Garden_Base**

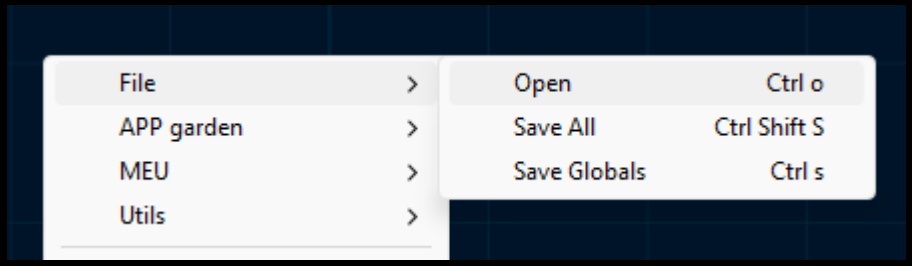
  Copy AAAAAPPs\Tuto\APP_Garden_Base

  Into **APPs_Guest**

  Rename APP_Garden_Base

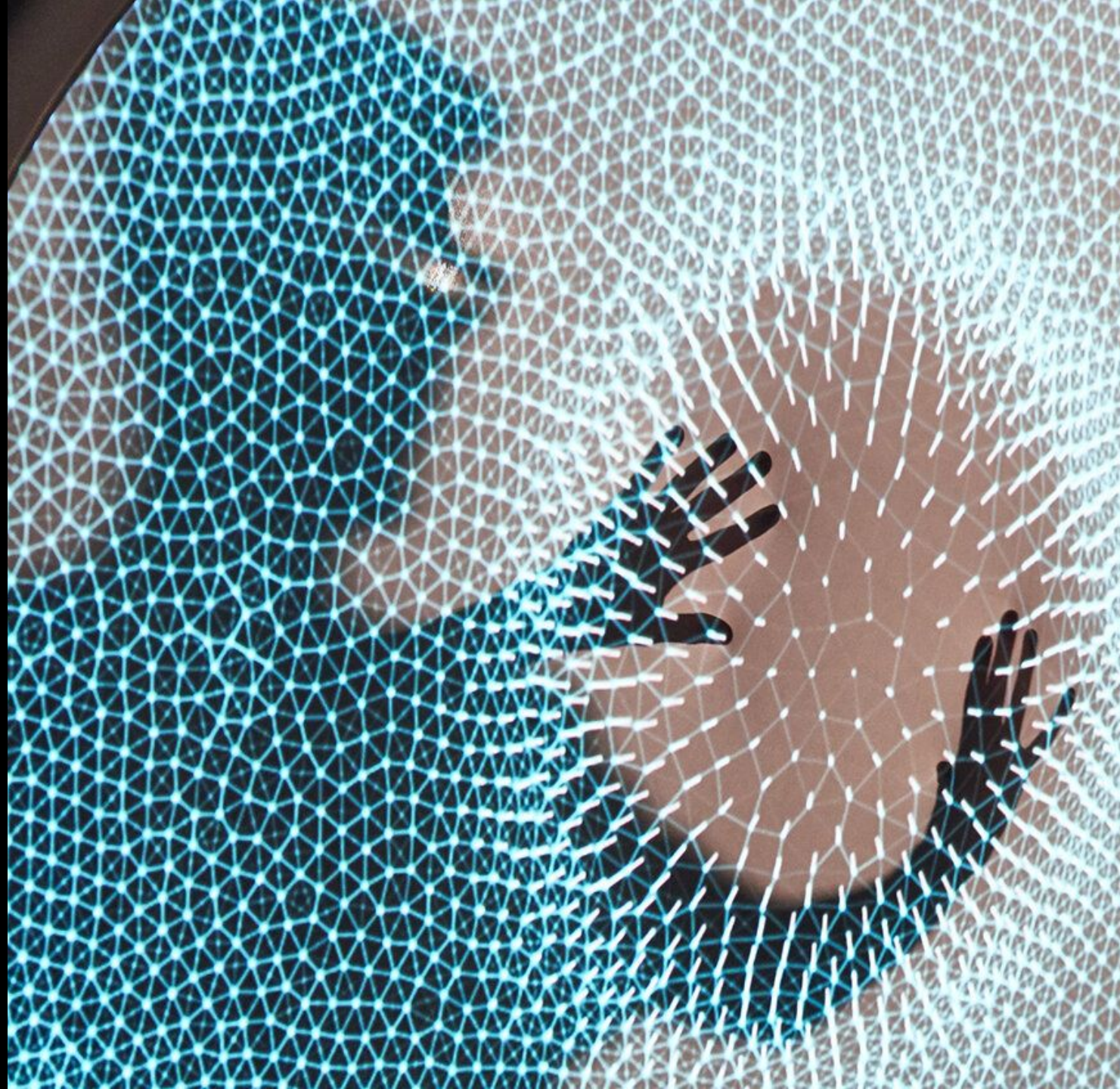
  Open AAASeed

  Open renamed **APP**

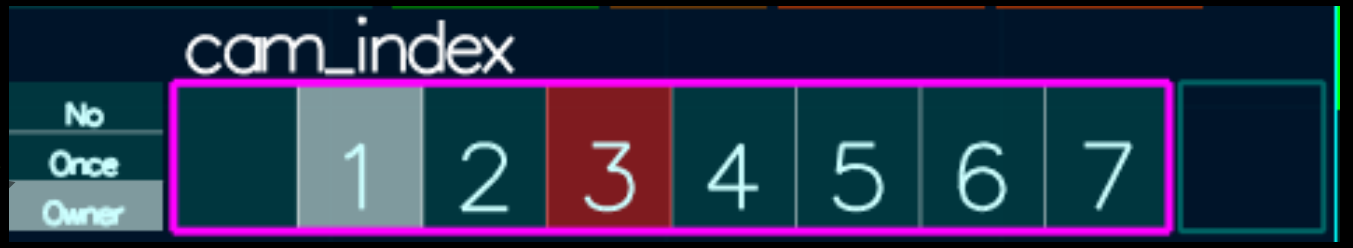


AAASeed
An introduction
Part 12:
Cameras & Axe

- MEU Cameras
- Camera Selector
- Axes Units
- Camera Edit
- Cameras Params



MEU Cameras



- **No / Once / Owner**
  - **No ⟶ No camera defined use current camera**
    the last one defined in the render chain (Owner see below)
  - **Once ⟶ use camera only for this MEU**
    then go back to current camera
  - **Owner ⟶ Own this camera**
    it becomes the current camera for this Meu
    and the next Meus in the render chain
- Camera selector
- Button **Axe show**

- Reminder: StarMenu ⟶ name

Camera Selector

- 8 cameras by default, 16 sometimes

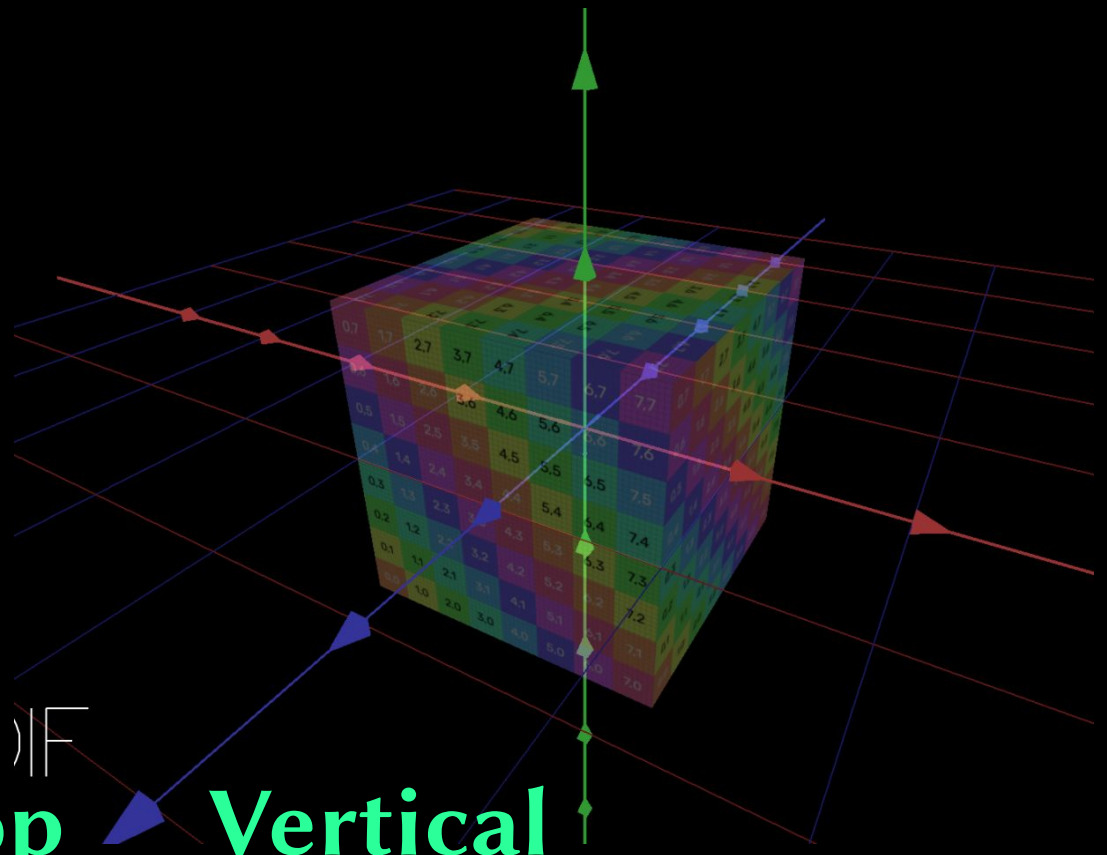
- Click — Select camera
- Ctrl Click — Copy the last camera clicked
- Ctrl C/Ctrl V — Copy / Paste
- Double Click — Edit in flatland
- Shift Click — Flip Lock / Unlock

  Locked camera are shown with a red background

  Locked camera can’t be edited with mouse and keyboard only using flatland.

Camera Axes Units

- **Right Hand** System
  like OpenGl
  not left hand
- **Z** in your **face**
- **2d : XY** facing you
- **X|Red** **left to right**
- **Y|Green** **bottom to top** **Vertical**
- **Z|Blue** **back to front**
- **No unit: Mathematics** not physics
  - Back of the pyramid on axes aligned with the unit



Camera Edit

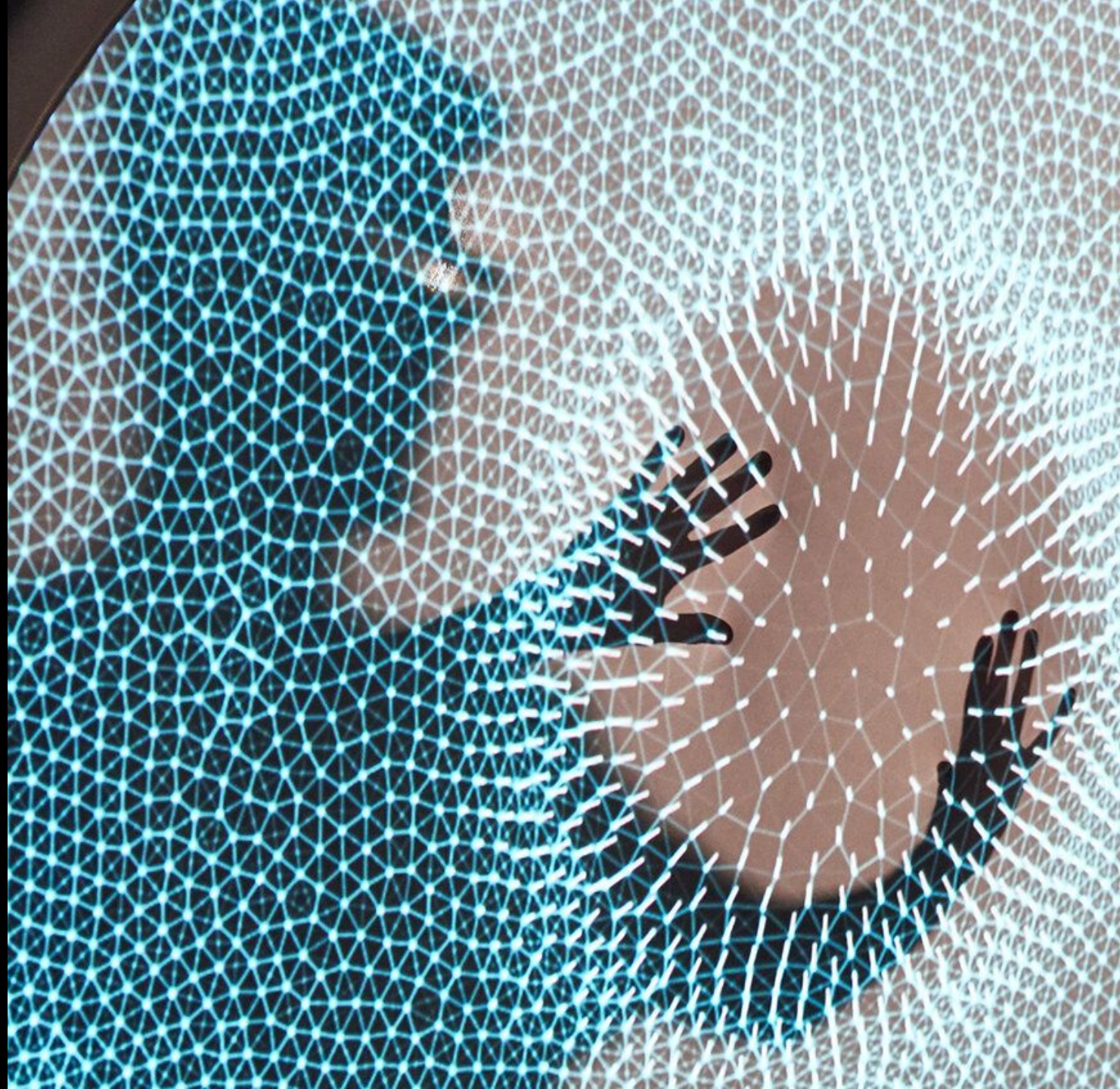
- **Middle Click → Start Editing**
  - BU_CAM will manifest if locked
    - BU_CAM itself locked
    - Or camera is locked
  - When editing
    - **Wheel** **Dolly in perspective, Size in orthogonal mode**
    - **Mouse** **Rotate**
    - **Middle Click Drag** **Move**
    - Key **o** **Flip (o)rthogonal vs perspective**
    - Key **x|X y|Y x|Z** **Align front | back**
    - **Ctrl Shift F4** **Reset**
    - **F4** **Focus** in **Flatland**
- **Left Click → accept edit**
- **Right Click → cancel edit:** go back to start position

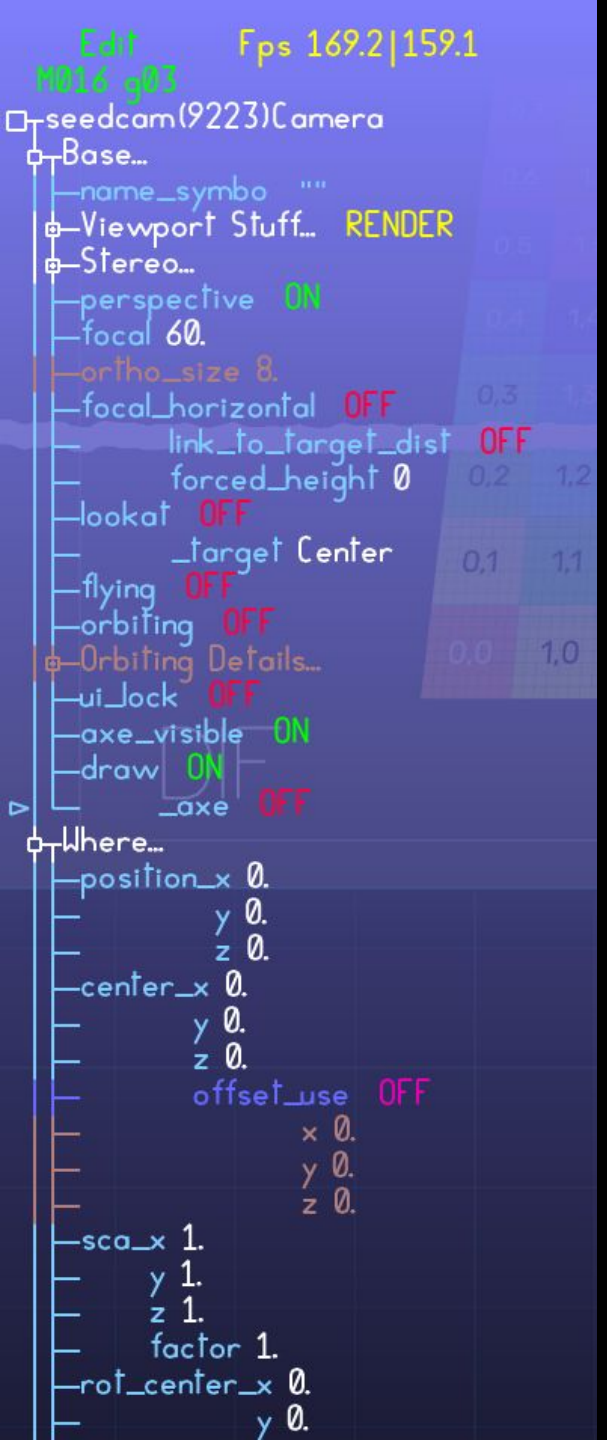
Camera Params

- To be added

AAASeed
An introduction
Part 13:
Flatland

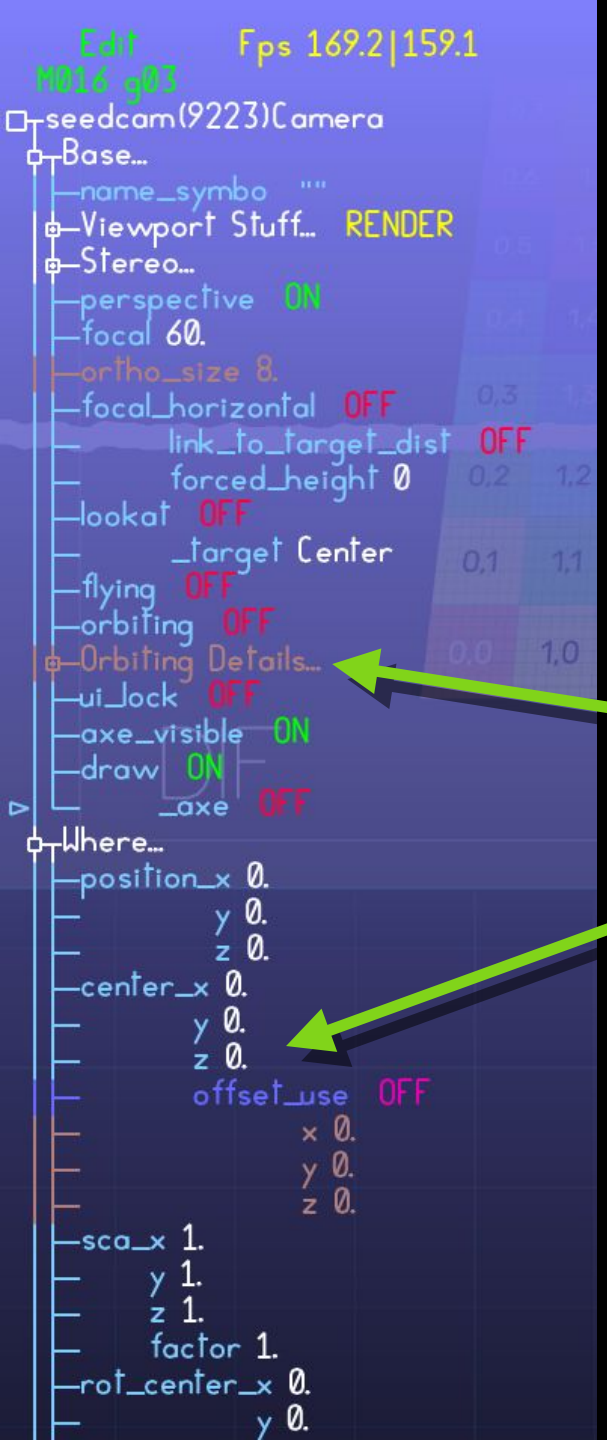
- Flatland
- Param
- Navigate Params
- Edit Param
  - **Keyboard**
  - **Mouse**
- Rotation in AAASeed





FlatLand a little deeper

- **Tree** Made of **params**
- **Tab** **On/Off**
  when flatland is displayed
  it get the keys event
  Not GaBuZoMeu
- **Focus (the current object shown in Flatland)**
  - **F10** **Preferences**
  - **Ctrl F10** **Start preferences**
  - **0** **Application**
  - **Triple n** **Network preferences**
- **Key <- ->** **Navigate** in past **Focus**
- **Wheel** **Scroll**



Param

Short for **parameter**

Param have Type:

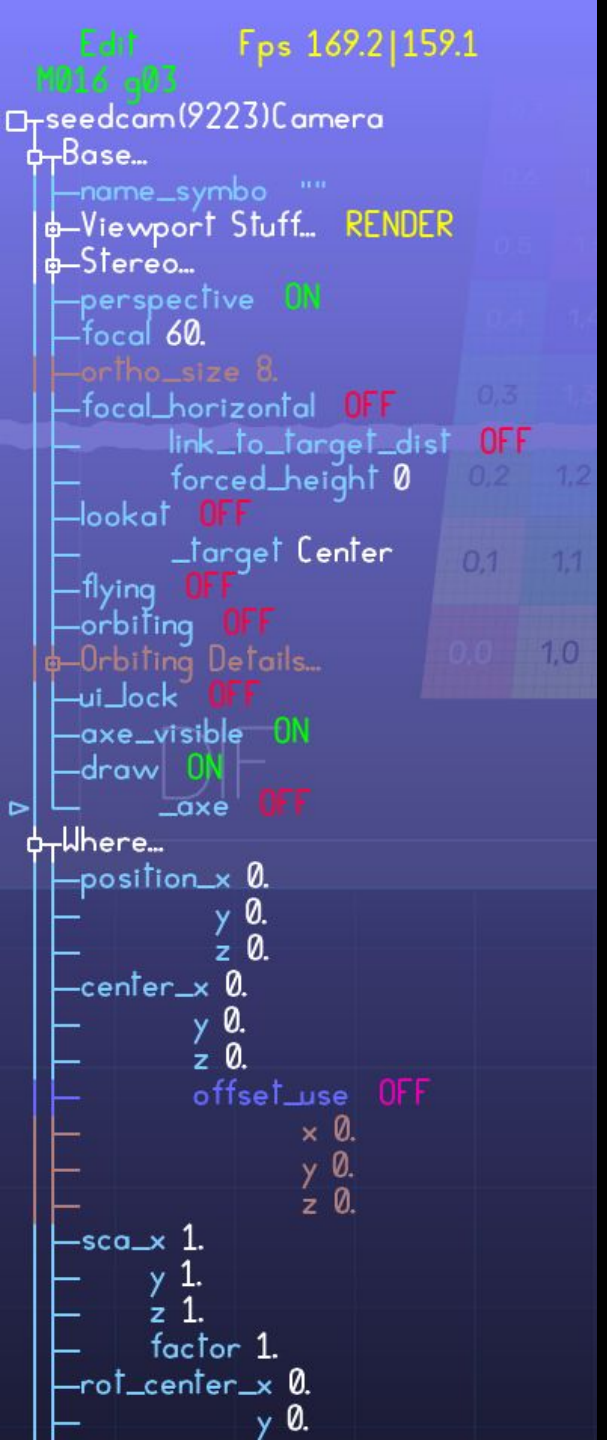
e.g. bool, integer, float, text, filename...

Param draw:

- **Box → something under**
  Big Box → Object
- **Param name**
  for C++ obj name is class_name(id)human_name
  Pinkish / Red param → Unused in this configuration
  no characters at beginning when same then previous line
- **Param value**
  for bool OFF ON
- **Yellow → Comment / Sum Up**

Param more info:

- **Big triangle pointing left just before name**
  parameter is generated by C obj it can't be edited
- **Small triangle pointing right at beginning**
  reference is defined



Navigate Params

- Click

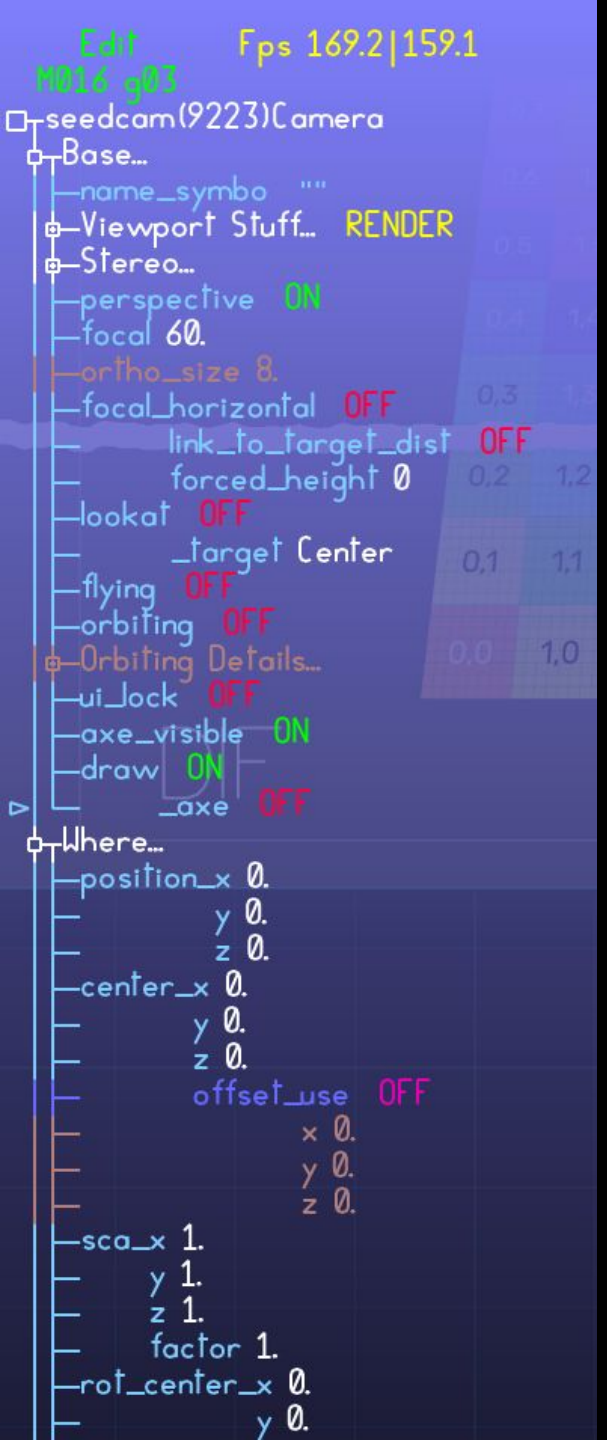
  Left — Select current param

  — eventually Open/Close

  Right — Menu

  — depends on position

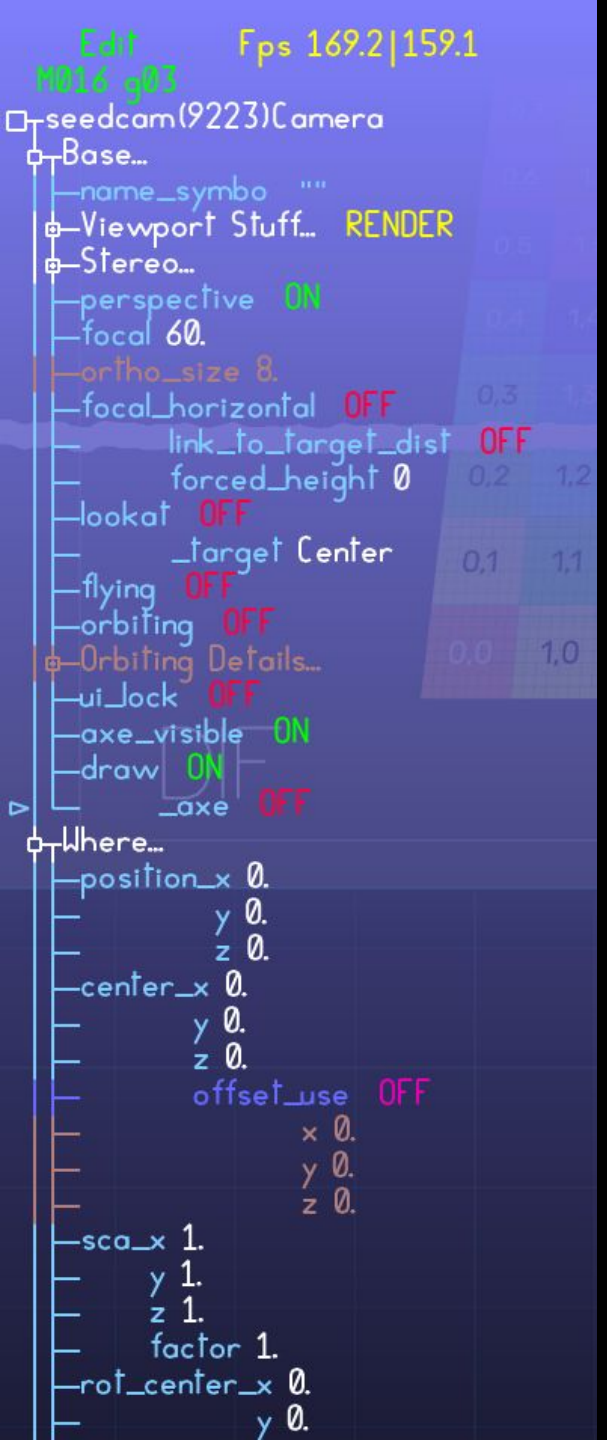
- Space — Open/Close
- Up Down — Change current param



Edit param: Keyboard

Similar to BU

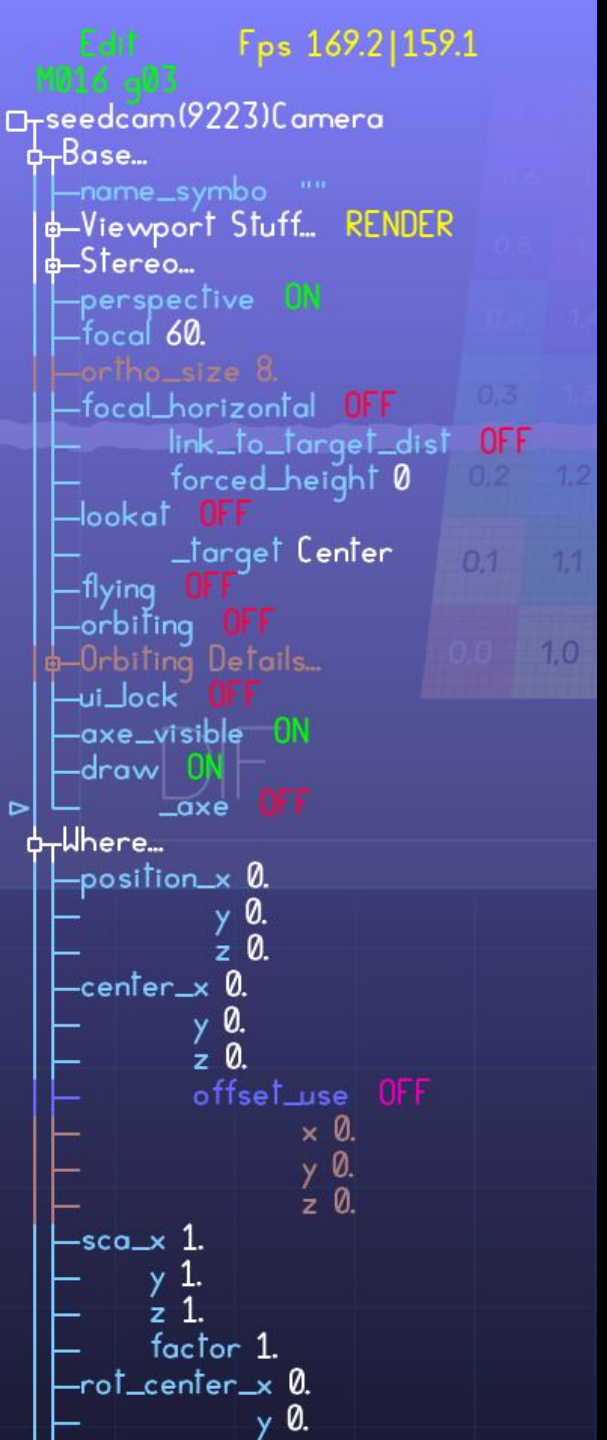
+ - * /	Change
Enter	Inverse
.	Floor integer
Home	Default
End	Inactive
PageUp	Maximum
PageDown	Minimum



Edit param: Mouse

- **Bool**
  - Drag left and Right
  - param_trig
    - Just click it
- **Integer, Float**
  - Hold click while turning around click point
    - clockwise to increase
    - counter clockwise to decrease
    - do as much turn you want
  - Change sensibility
    - Ctrl Slower
    - Ctrl Tab Even Slower
    - Shift Ctrl Faster
    - Shift Ctrl Tab Even Faster
- **Double Click**
  - Dialog or trigger action

- **Developer note**
  - Alt Middle Click → flip debug view for param



Rotation in AAASeed

- Unit is Turn / Revolution

  No radians, degrees or gradians.
- 1 full turn
- 0.5 half turn
- 0.25 quarter turn
- –0.25 quarter turn reverse
- Canonical aspect

  [0.,1.] map full circle

AAASeed
An introduction
Part 14:
Texture

- Image and texture
- CPU and GPU Memory
- Texture channels
- Channel type
- Banks of Binds:

  Texture Slots
- BU_BANK
- BU_BANK Edit
- BU_MONITOR StarMenu

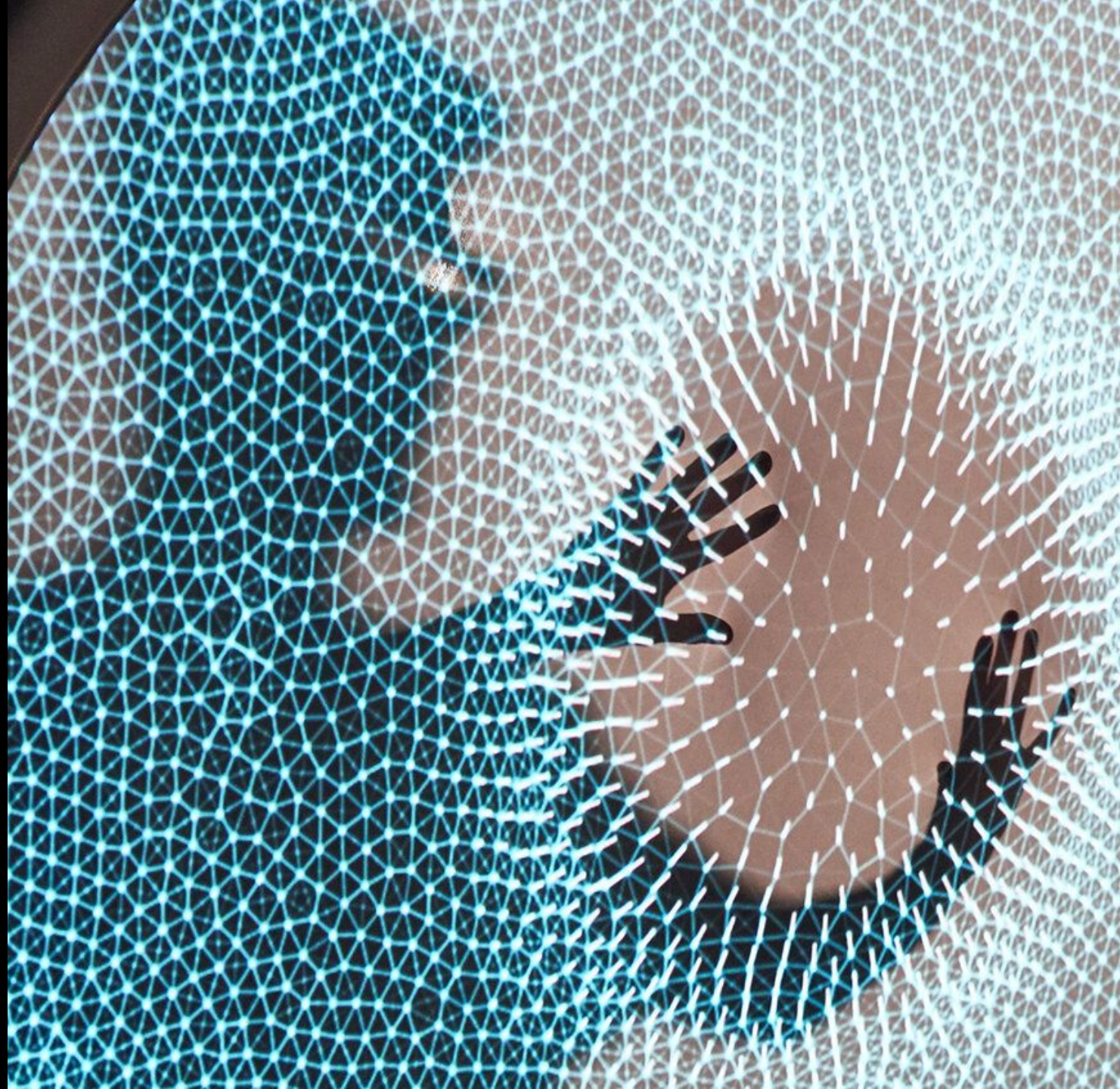


Image and **Texture**

- **Image**

  2d grid of pixel

  On the CPU side

  CPU mean **Central Processing Unit**

  The Intel or AMD processor used on your computer

  CPU side mean **CPU Memory**

  e.g, when a jpg file is read it is first moved to CPU memory
  then decoded by the CPU and stored there.

- **Texture**

  Same as an image but on the GPU side

  GPU mean **Graphics Processing Unit**

  The massively parallel Graphics processor on your computer
  Nvidia or AMD graphics card

  or a less powerful « integrated Graphic Card »

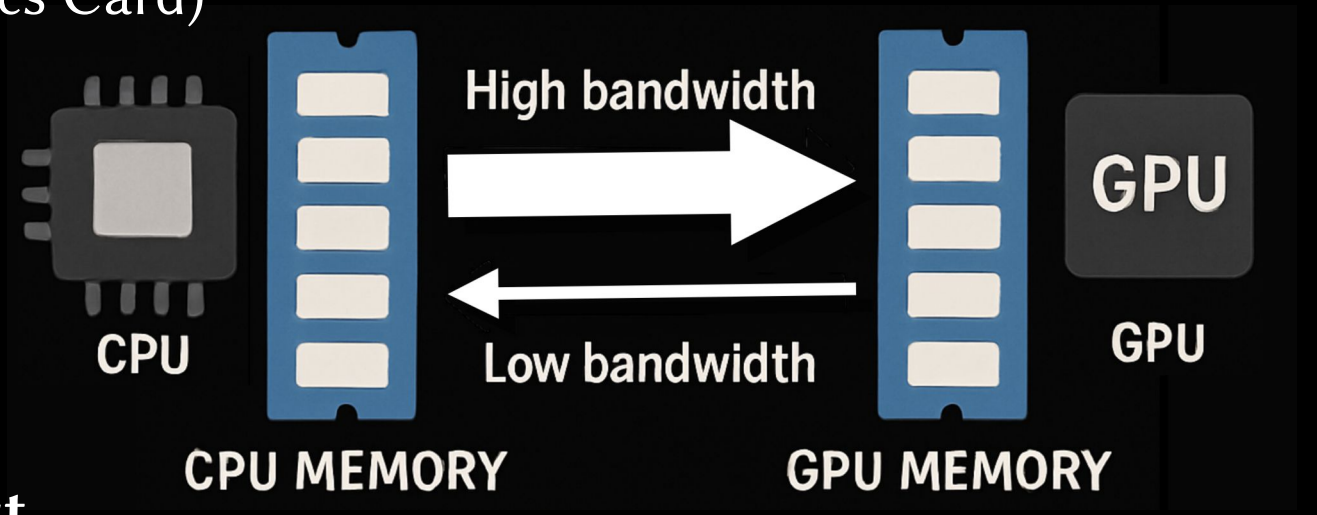
  e.g. the image decoded from from a jpg file stored on the CPU side is moved to the GPU side so it can be drawn by the GPU

- In fact we want you to understand that there is two different memory and that data have to be in the right memory to used by the ad hoc processor.

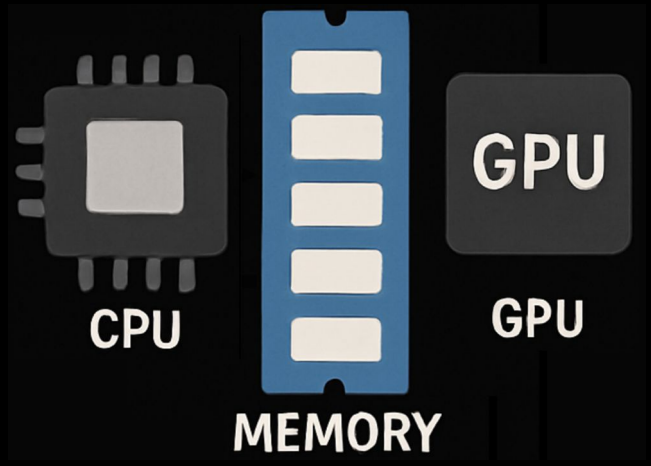
CPU and **GPU Memory**

- **Separate Memory** (Discrete Graphics Card)
  - **Memory Transfer to do**
    CPU to GPU transfer is fast
    GPU to CPU transfer is slower
    - **can cost a lot**
      developer note
      - it stall the pipeline

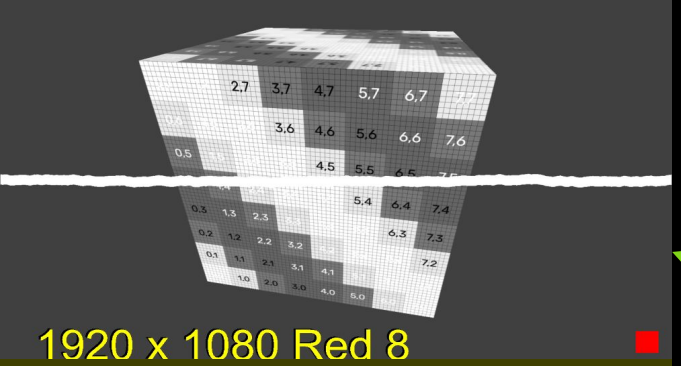
    GPU memory can be really fast

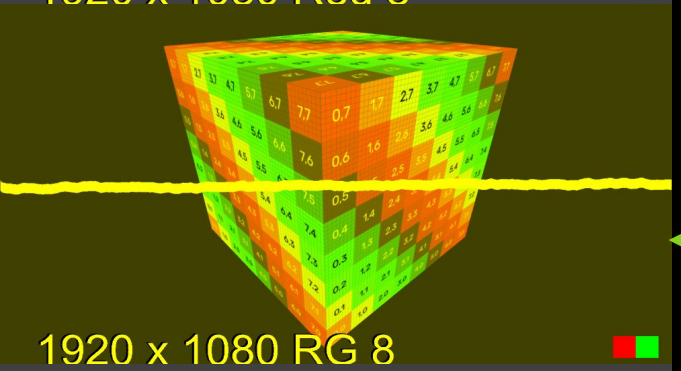


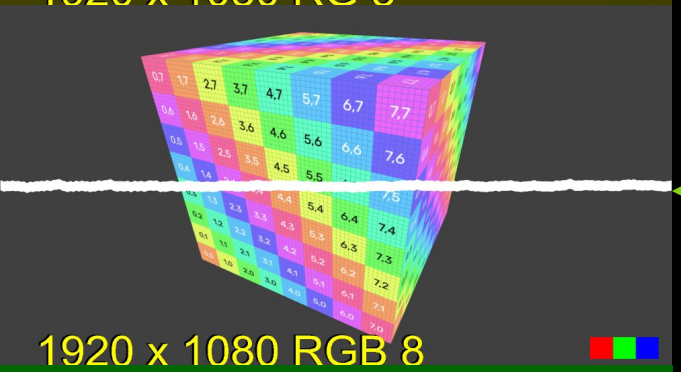
- **Unified Memory** (Integrated Graphics)
  - **No memory transfer**
    Memory is slower
    - than dedicated GPU memory

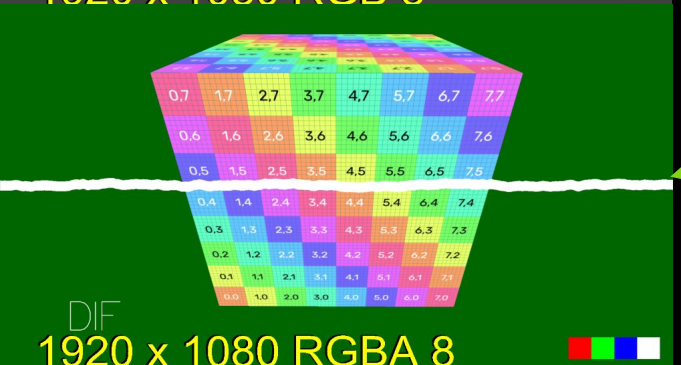


Texture channels









See BU_MONITOR part

- **1 or R for Red**
  showed as **greyscale**
- **2 or RG for Red + Green**
  showed as **yellow**
  no Blue
- **3 or RGB for Red + Green + Blue**
  full color
- **4 or RGBA for RGB + Alpha**
  like RGB but with transparency/Opacity
  0 mean transparent, 1 Opaque

Channel type

- AAASeed support texture where **all channel (component) have the same type**
  - **Integer 8 bits (8) → 256 values, from 0. to 1., about 2 decimal digits precision**
    Compact, used if no more requirements.
  - **Integer 16 bits (16) → 65 536 values, from 0. to 1.,about 5 decimal digits precision**
    Use when more precision is needed (e.g. HDR).
  - **Float 16 bits (16fp)→ from –65 536. to 65 536., about 3 decimal digits precision**
    Also called half for half precision floating point
    Fix floating point format,
    Use when 0. to 1. is not enough.
  - **Float 32 bits (32fp)→ wide range, about 7 decimal digits precision**
    Fix floating point format,
    Use when its precision or range is needed.
- **Smaller is faster**

- AAASeed support others channel type
  e.g. Depth/Z buffer 24bits, compressed format, YUV ...
  but you can ignore this for the moment

Banks of Binds : 2048 Banks of 32 Binds
⟶ 65 536 Texture Slots

- Texture are referenced at low level from 0 to 65 535
  - we say flat bind
  - Texture are referenced most of the time with
    - a bank index [0, 2047] and a bind index [0,31]
    - flat bind = bank * 32 + bind
  - Some Texture are referenced by a texture name
    - more in Part Texture More

- Each slot can store a file name
  - when a bind/slot is accessed if there is no texture and a file name is defined AAASeed try to load in the slot.
  - This is how you already have certain texture when you start AAASeed
  - Flatland then key 0 to focus on app/Bind_list/Image/bind
    - if you want to see the list of filename
  - Save as global in User Pref (Double Esc or Ctrl S) with a .image_bind extension

BU_BANK

BANK

No texture

Slow blinking redish

Blue diagonal

Each bind a Monitor:

Click Double or Space

Open monitor

StarMenu Extended

Bank selector

bank contains 32 binds

Bind in bank

[0,31]

Name

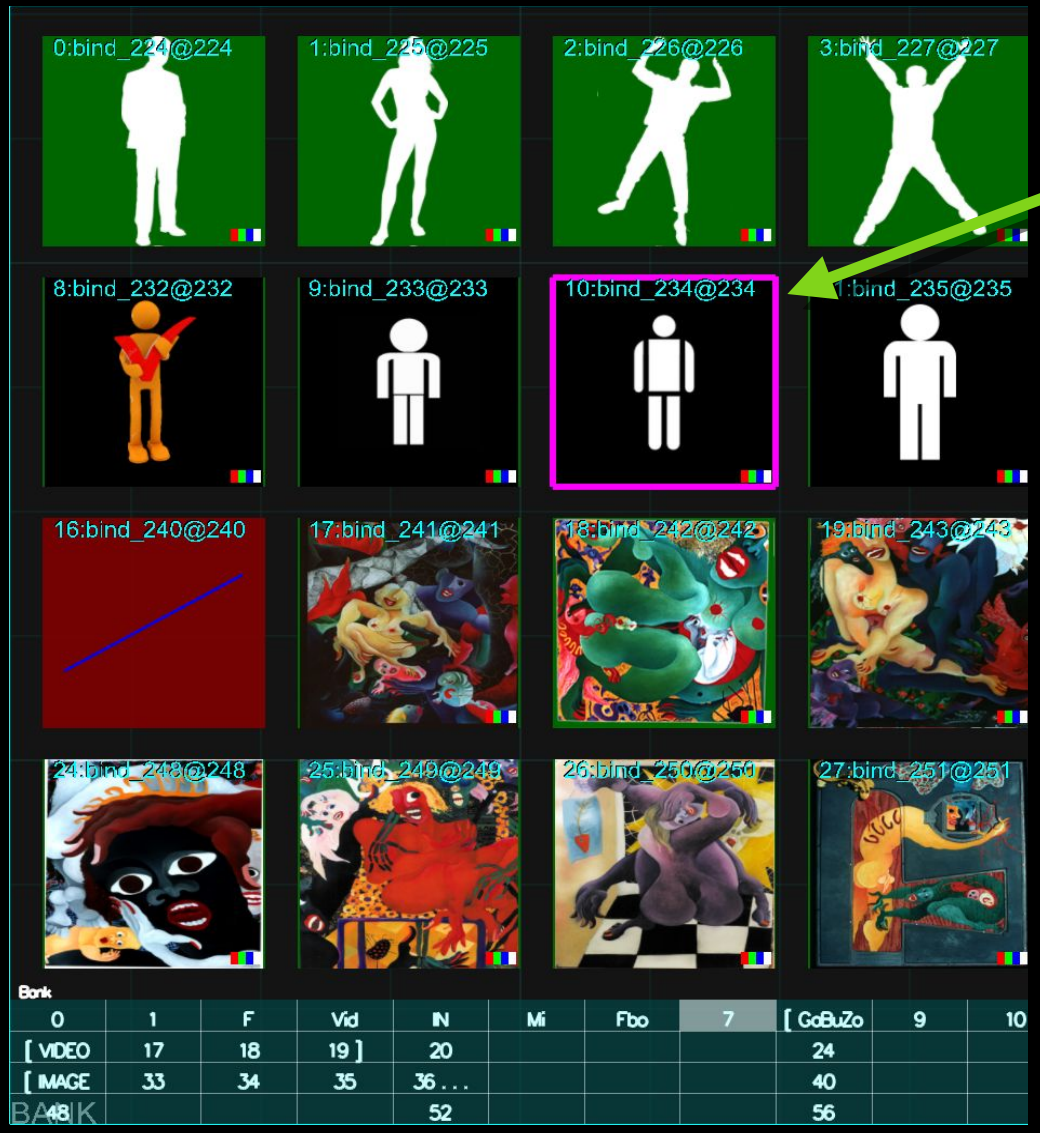
Flat bind

[0,65535]

bank offset

0 to 1984

BU_BANK Edit



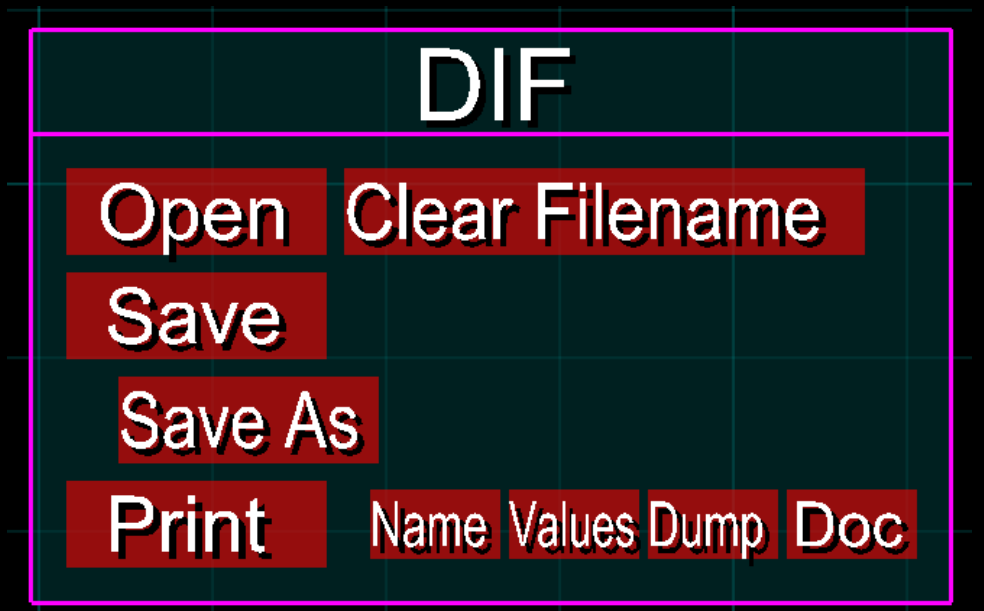
- When a BU_BANK monitor is the current BU
  and flatland is off (reminder)

- Ctrl Arrow
  Swap texture and filename with the bind in the arrow direction
  Stay in the current bank.

- Ctrl C/Ctrl V
  Copy/Paste texture and filename
  Can be done accros bank

BU_MONITOR StarMenu reminder

- **Click and drag out quickly (Quick drag-out** like any BU where it doesn't appear directly)
- **Open**
  - **Load an image via a file dialog**
    - **at the current bind**
    - **can choose several files**
      - loaded in successive binds
  - **Set the the filename for this bind**
- **Clear Filename**
  - **Clear the the filename for this bind**
- **Save**
  - **Save the Image to** Media/AAASnapshots
  - How it is saved in Flatland Prefs/Image Save
  - **Note: Save do not use the bind filename set by Open**
    - this is not a bug but a feature
- **Save As**
  - **Save with a file dialog to choose location**
- **Print**
  - **Print the image**
  - How it is printed in Flatland Prefs/MASTER/Print



BU_MONITOR BU_BANK StarMenu

- **Quick drag-out** like

- **Erase**

  Destroy the image in this bind

  Clear the filename too

- **Dec**

  (Dec)rement the texture index

  in fact swap with the previous one

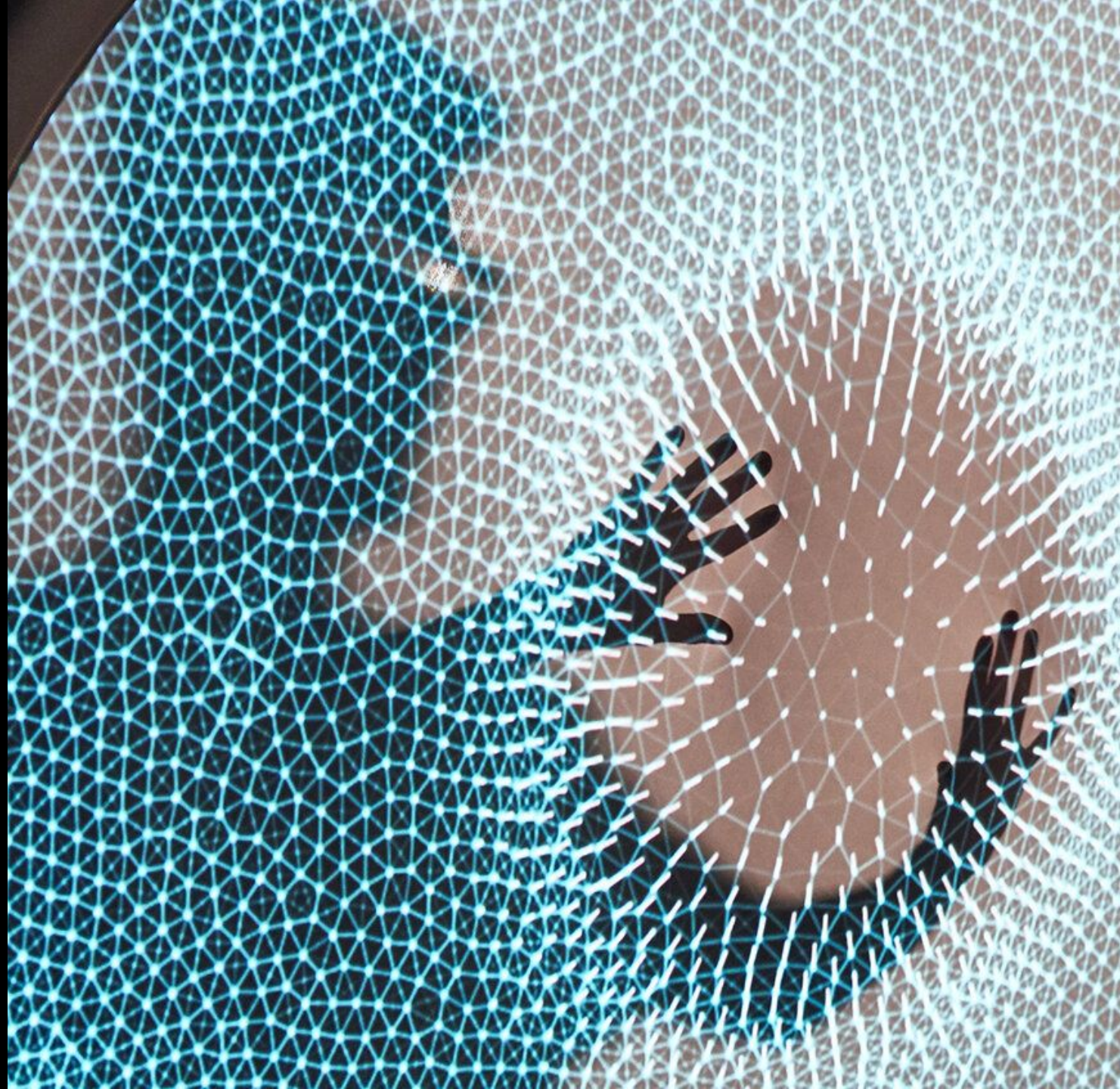
- **Inc**

  (Inc)rement the texture index

  in fact swap with the next one

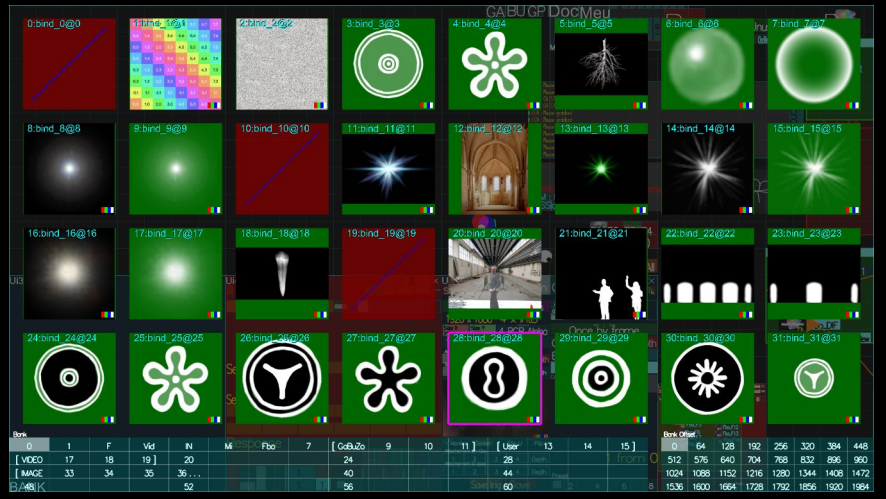
AAASeed
An introduction
Part 15:
Texture More

- An apology
- BU_BANK and BU_TEXTURE
- Texture Name
  - **-More**
- BU_TEXTURE
  - **Shortcut**



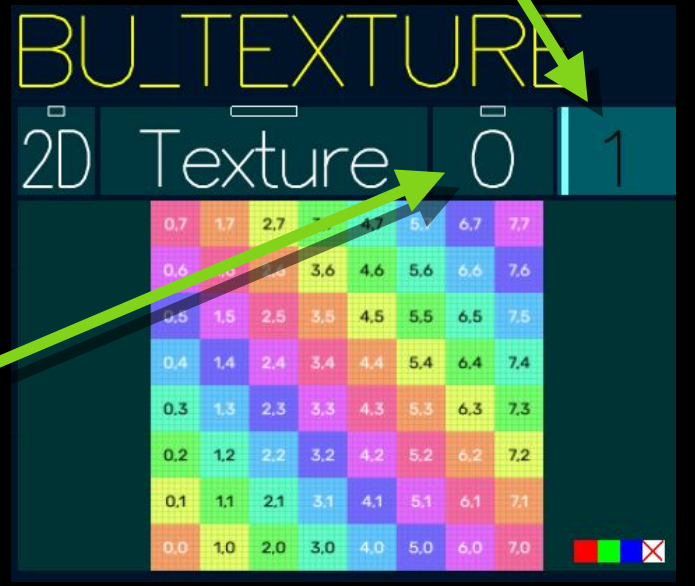
Disclaimer on current texture architecture

- It is a problem
  - Mâa apologize deeply
- We now it, it is tricky
  - Bank fixed size
  - No User Bank
  - ...
- We could not change it before the end of the ArtCast4d project
  - There will be MEU Bank to define and edit them
  - Future Banks will have names (not only number) and each their own bind size
- We have to live with it for a while (2025)
  - This is stable and solid.
  - The new texture architecture will probably keep the current structure at the bottom and build on top of it.



BU_BANK and BU_TEXTURE

bind



bank

Texture Name

- **Bank and bind was too abstract**

  Texture Name concept was added.
- **Each App have a set of names**

  corresponding each to a flat bind index

  a name → a flat bind → names

  e.g names A to P used in general for 16 Videos slots.
- **Ui in particular BU_MONITOR use and display name when they exist.**
- **Names are defined case sensitive (for a nice display) but search is case insensitive.**
- **A name is eventually associated with**

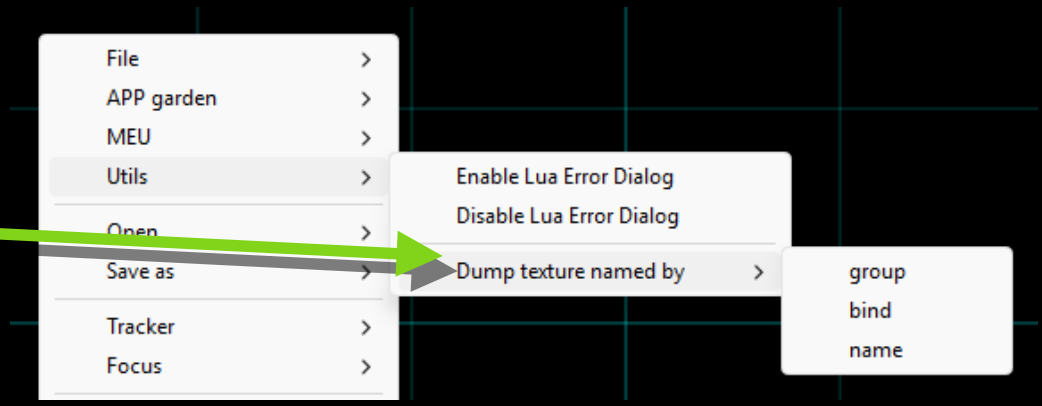
  a short name

  a synonym.

  any can be used.
- **Texture name are regrouped in groups.**
- **One texture can be referenced by several names.**

Texture Name more

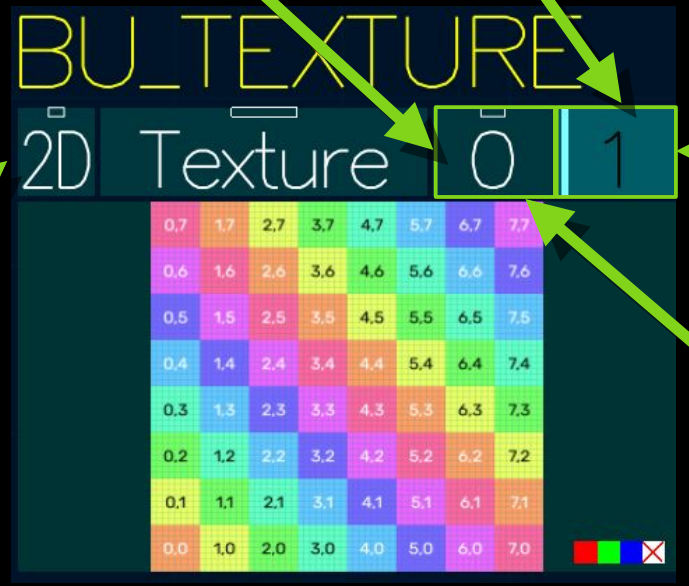
- If you want to see all the texture names used in an App
  use the Utils SubMenu
  All names will be printed in the terminal window.
- Main groups
  - F1 to F24 24 binds from bank 2 (or F)
    used by MEU_Video in general
  - A to P 16 binds from bank 3 (or Vid)
    used by MEU_Fbo in general
  - In1 to In32 32 binds of bank 4 (or IN)
    used as Input in general
  - Mi1 to Mi16 16 binds from bank 5 (or Mi)
    used by MEU_Mire in general (Pattern in french)
  - M1 to M32 32 binds of bank 11
    used for Monitors
  - A pass group use for multipass rendering
    this will be treated in separate document
    e.g. Albedo, Normal, Specular, Emissive, Light, Fxaa ...
- A lot of these names appear in the shortcut menu in the next slide
- A texture name DIF is defined
  - This bind is often use by default to draw in (see FBO part)

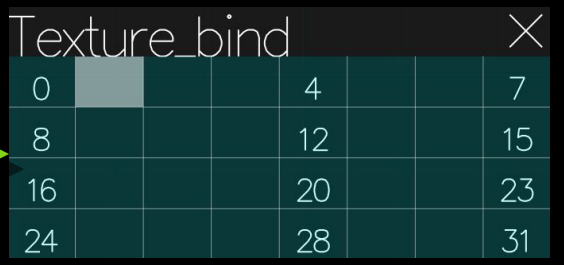


- **BU_TEXTURE choose a texture**
  - A bank BU to choose the bank

    click open a selector
  - A slider to choose the bind

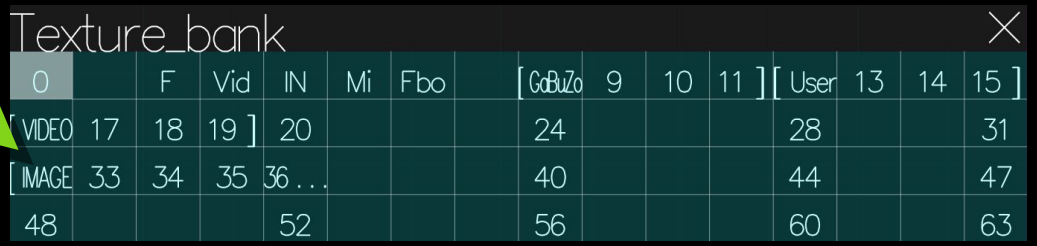
    click open a selector and starMenu





Dimension
(Optional)
up to the MEU

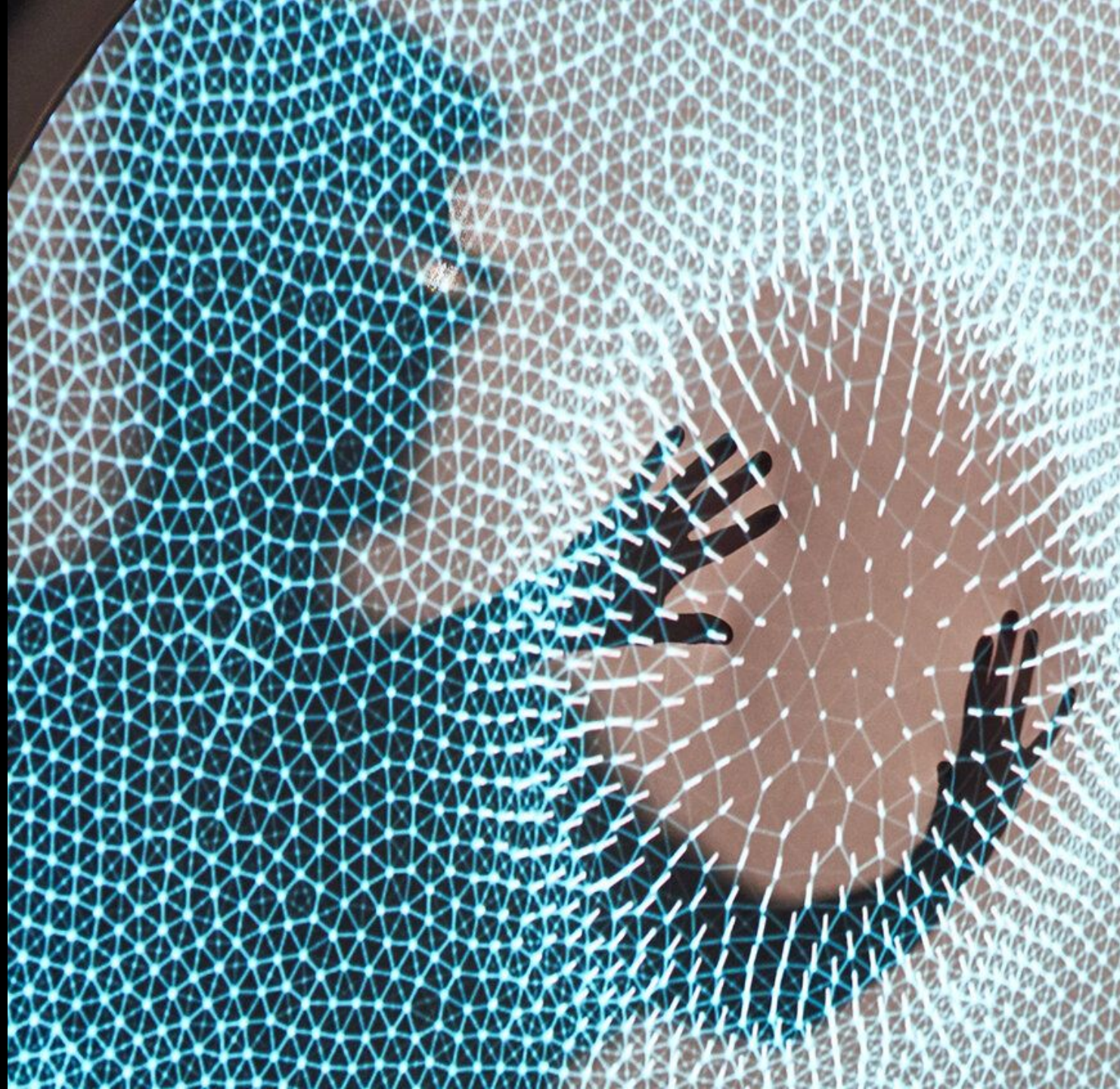
Double Click in these to Load texture in bind



- **BU_TEXTURE Shortcut Menu**
  - A shortcut Menu simplify the manipulation
    click open a selector with the most used texture name
    bank and bind are change in accordance
  - Monitor can be used here too (Click zones, StarMenu, Keyboard)

AAASeed
An introduction
Part 16:
Fbo

- Frame Buffer
  - Canvas
- Fbo/Frame buffer object
- MEU Fbo and render chain
- MEU Fbo
- MEU Monitor



Frame Buffer

- Concept from **Computer Graphics and Video**
- **Frame = Image**

  A frame is a single complete image in a sequence—like a frame in a video.
- **A framebuffer is a memory buffer** that stores the color values of each pixel before they are displayed on the screen.

- Think of it as a **digital canvas**

  The GPU draws into this canvas. It's where rendering results are stored—either for display or further processing.

  where the pixel are written and stored to be displayed or reused later
- **Used in Effects and Multi-Pass Rendering**, Framebuffers aren't just for final output:

  Effects like shadows, reflections, bloom, and motion blur often render to intermediate framebuffers.

  These act as scratchpads—temporary storage used to build up complex visual effects over multiple rendering passes.

Fbo / Frame buffer object

- In AAASeed Frame Buffers are Fbos
- Fbos regroups textures (image) of the same size called attachments.

  - Color attachments

    we use 1 most of the time, but AAASeed deal with up to 4 if neeeded.

  -Depth attachment (used to do z-Bufferring)
- Fbo are needed to draw in textures.

- AAASeed use a MEU_Fbo

  Special MU display

  Icon changing size

  Fbo yellow letters flashing

Meu Fbo and render chain

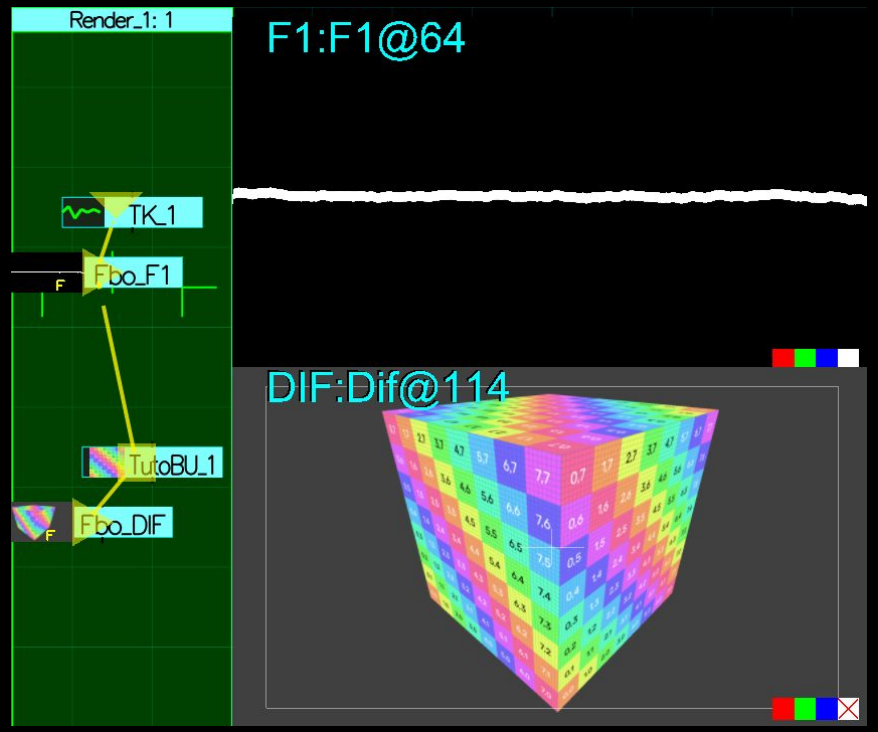
- No Fbo

  we draw to the back of the window, under the Ui.

- A Fbo

  All the successive Meu in the rendering chain use it

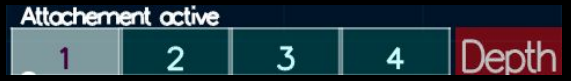
  until a new one is executed/rendered



MEU Fbo

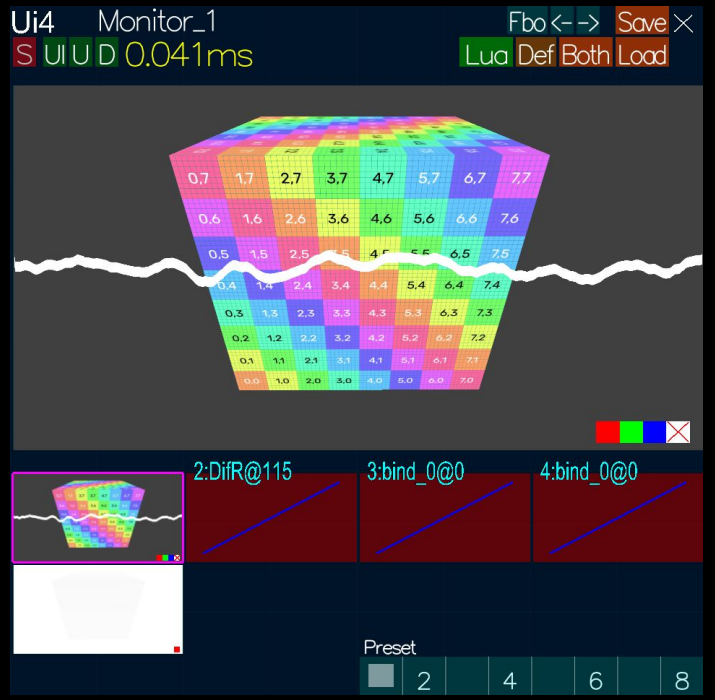
- **Size**
- **Color format**
  - **Channel number** R RG RGB RGBA
  - **Chanel Format**
    - Integer 8 bits [0,255]
    - Integer 16 bits [0,65535]
    - Floating point 16 bits, called half
    - Floating point 32 bits

- **Attachments**

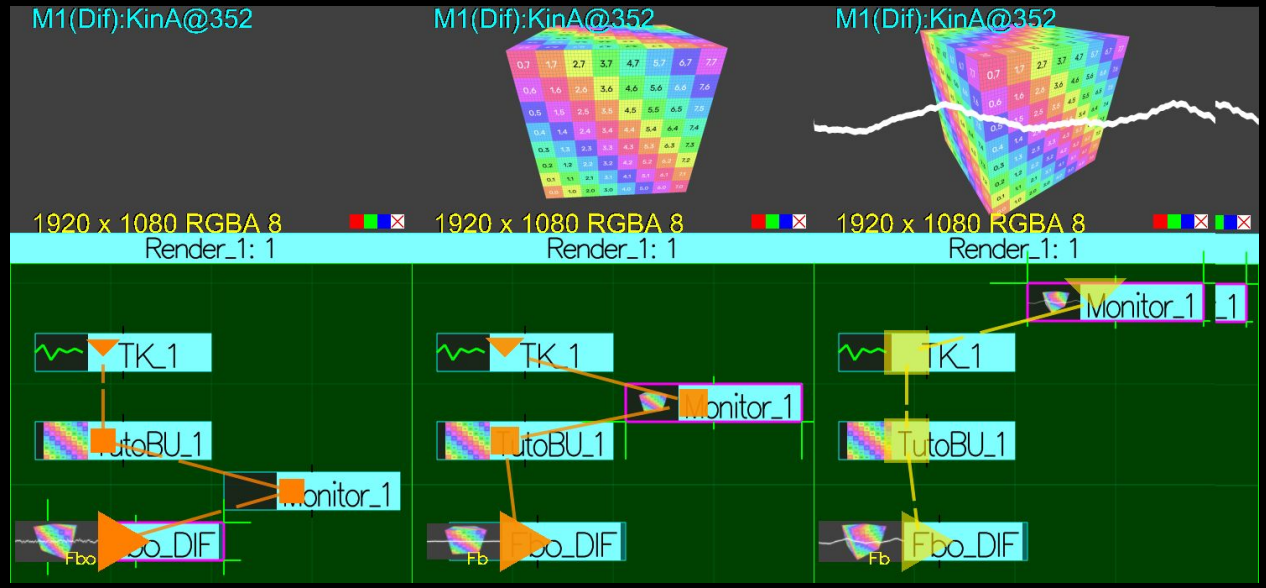


  - **Selection**
  - **Cpu** : transfer from Gpu to Cpu
  - **Monitor**
    - **Active**
    - **Click Double**
  - **Save Trig → One frame**
  - **Save → Each frame**
- **Cameras**
- **Erase**
  - **How, What**
  - **Colors with presets**
  - **Trail**

MEU Monitor



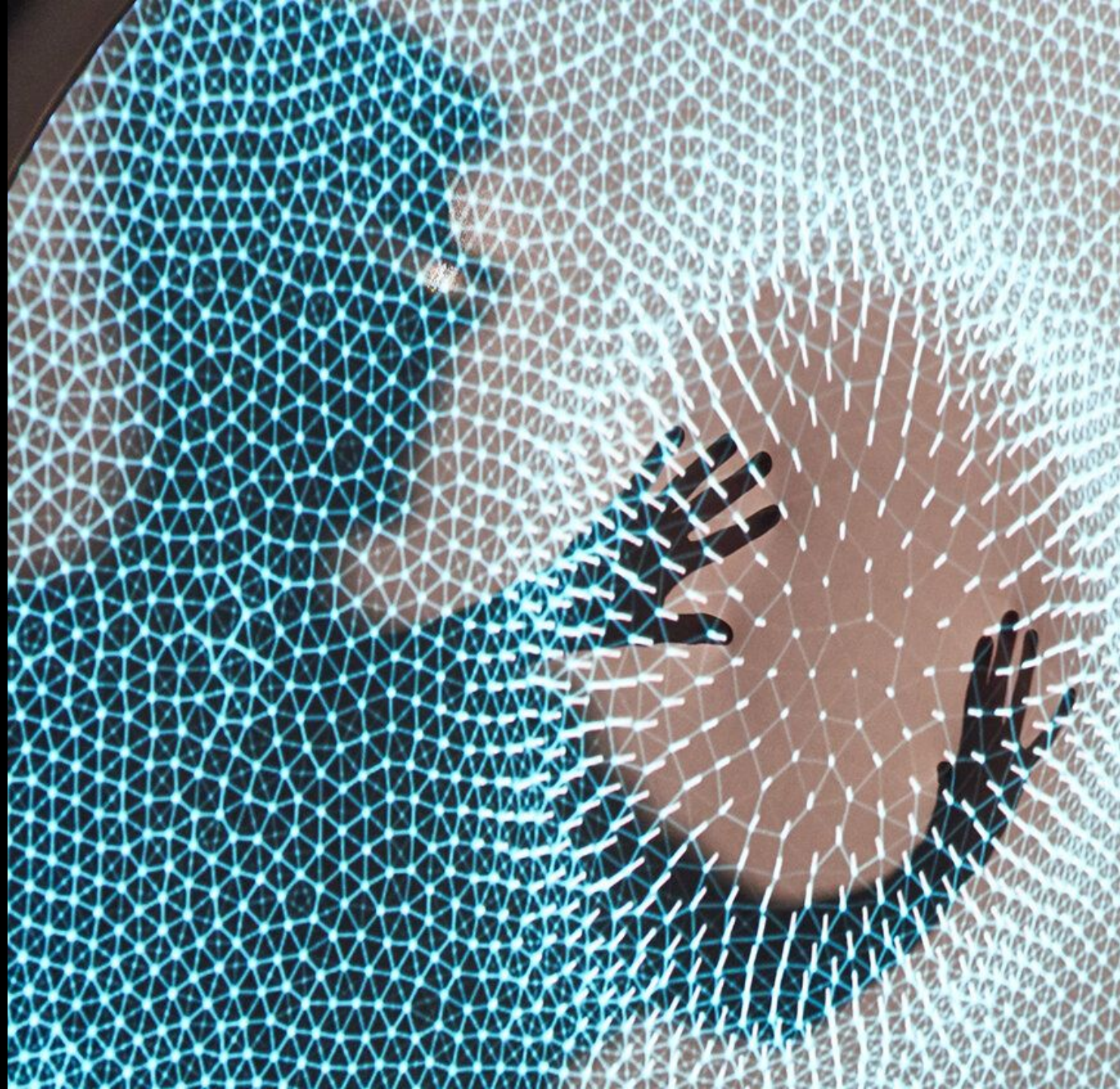
- **Copy** and **Display** current **state** of current **Fbo**
- **Monitor** show selected attachement



- **Meu**
  - **Show all attachements**
  - **Select monitor attachement**

AAASeed
An introduction
Part 17:
Create MEU

- Core MEUs
- Fun MEUs
- Create MEU
- Import MEU



Core MEUs

- **Fbo** Frame Buffer Object
  - Canvas where we draw
- **Monitor**
  - Display and copy the current state of the current Fbo
- **Dir**
  - Encapsulate MEUs
- **Video**
  - Decompress video in a texture
- **PIP** Picture In Picture
  - Display a texture
- **Out**
  - Ouput a Texture
- **TRAX**
  - Plug BU
- Kinect
- Analyse

Fun MEUs

- **TK**
  - Visualize the Audio
- **Displace**
  - Displacement mapping
- GeoUV
- Obj3d
- Boid
- Deform like class
- Lua Draw
- ShaMosaic

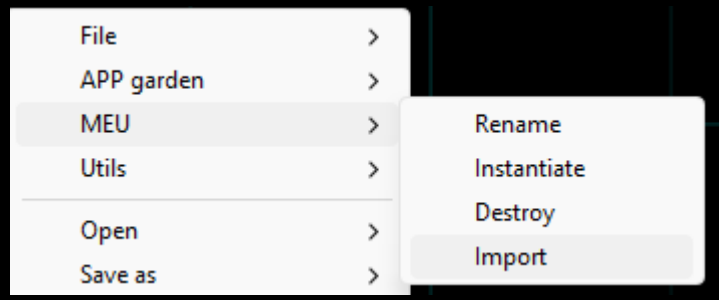
Meu Window: Create MEU

- Navigate MEU Prototype
- Create New MEU
- Select by Tag
- Select by Name

Meu Window: Import MEU

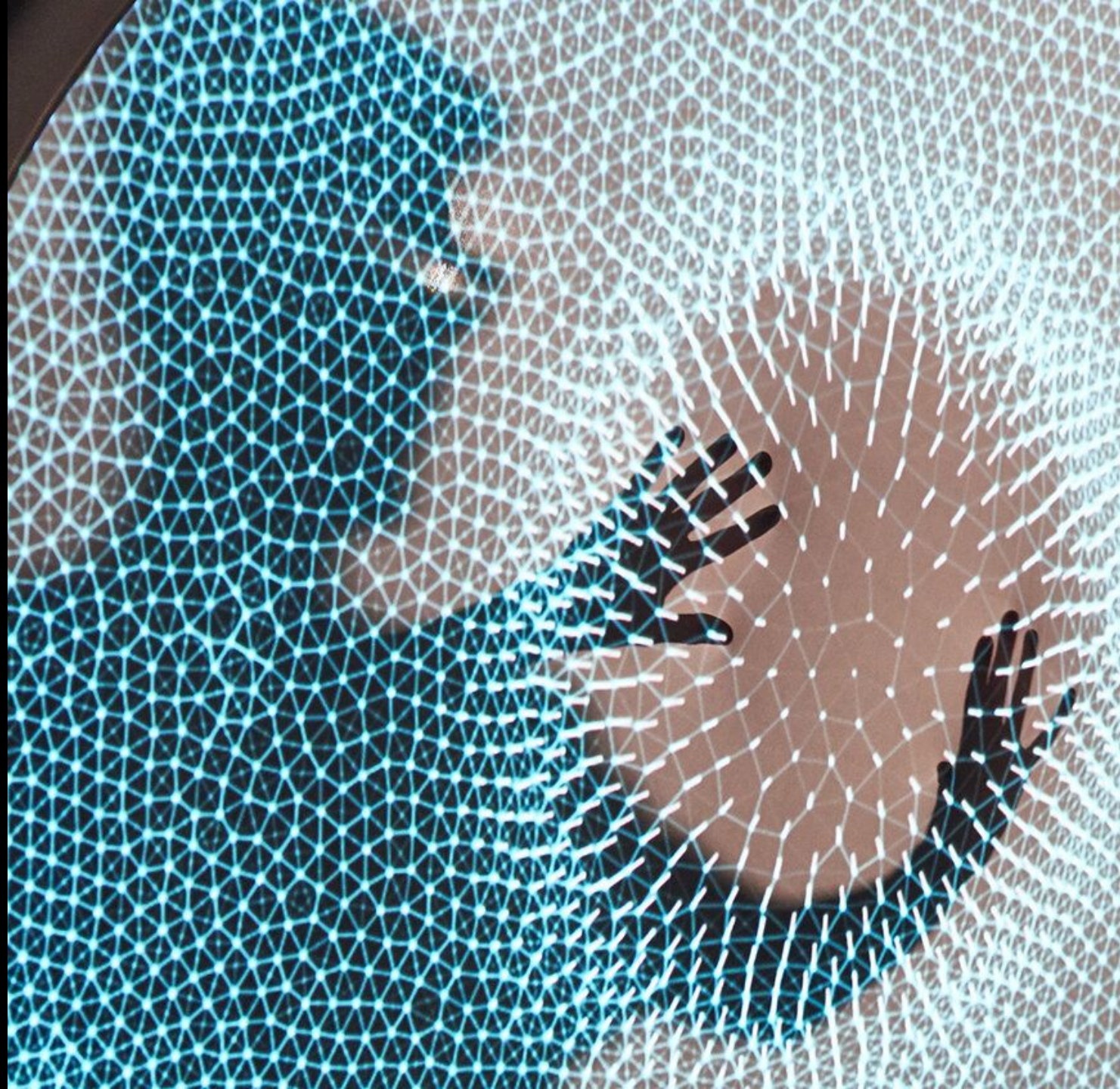
- Menu File/ Import MEU
- Navigate to an APP
- Navigate to a MEU

  folder AAAMEU
- Open

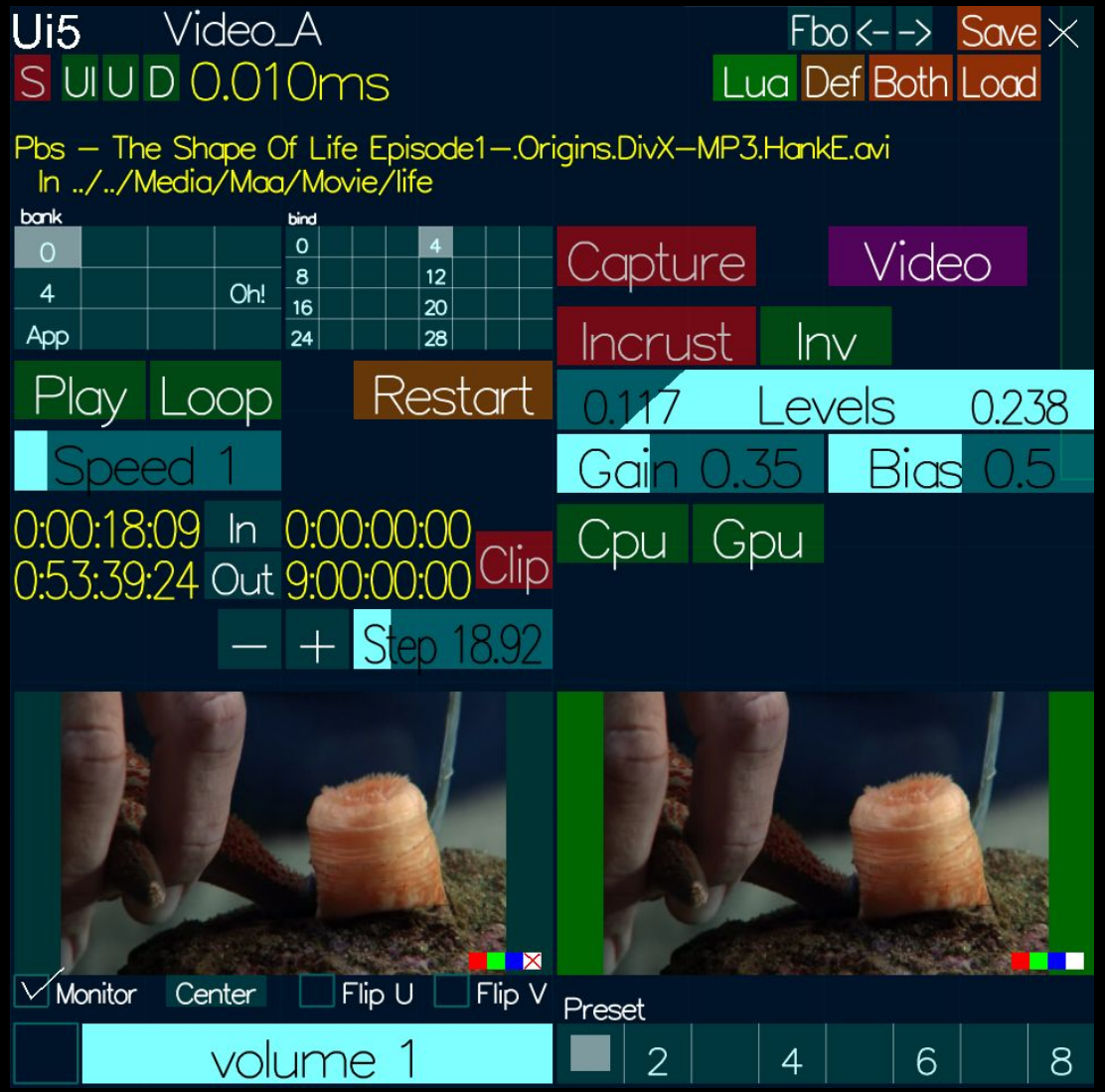


AAASeed
An introduction
Part 18
Reading video

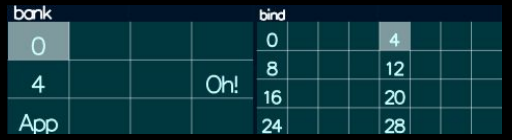
- MEU Video
- Gain Bias



MEU Video: Send an Image stream to a texture



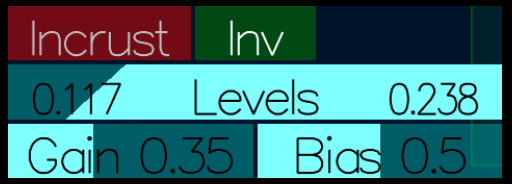
- **Video Name and folder (Red when a problem)**
- **Bank / Bind like texture**
  - **2 selectors**
  - **Click double**
    - **Pick a video for the bind**



- **Play / Control**
  - **In / Out / Clip → play a subpart**
  - **- / + → move in time of Step**

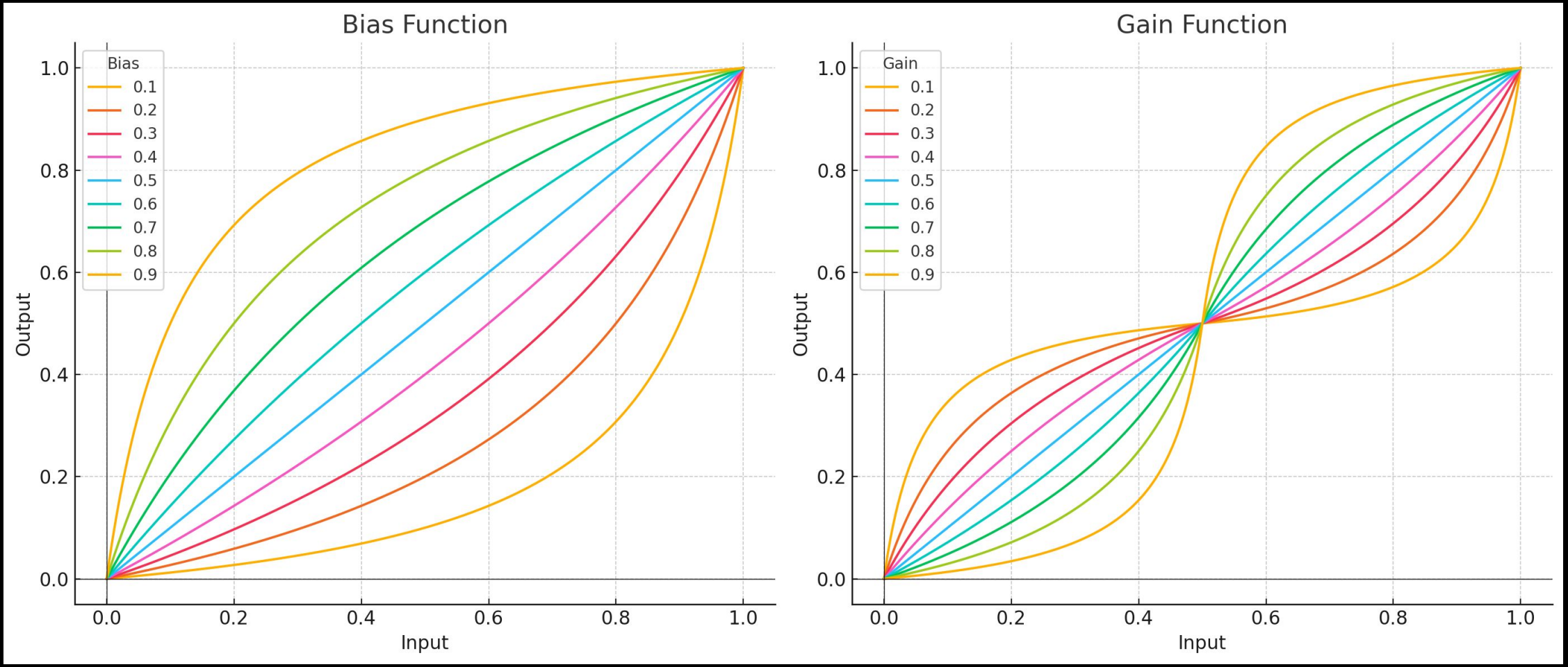
- **Monitor**
  - **Click double open as separate**
- **Volume**

  - **Check button** with **Slider**
  - There is a master Volume Too
- **Capture → switch to capture mode**
  - more in a separate document
- **Video → Flatland** (Violet)
- **Incrustation**
  - **Luminance Incrustation**
  - **Inv**erse → inverse the curve
  - **Levels** → min max levels of inscrustaion
  - **Gain / Bias** alter the shape of the transfer curve (more next slide)



- **Cpu / Gpu**
  - **Decompression** happens in the **CPU** Side
  - **Where we keep it**

Gain Bias an electronic heritage

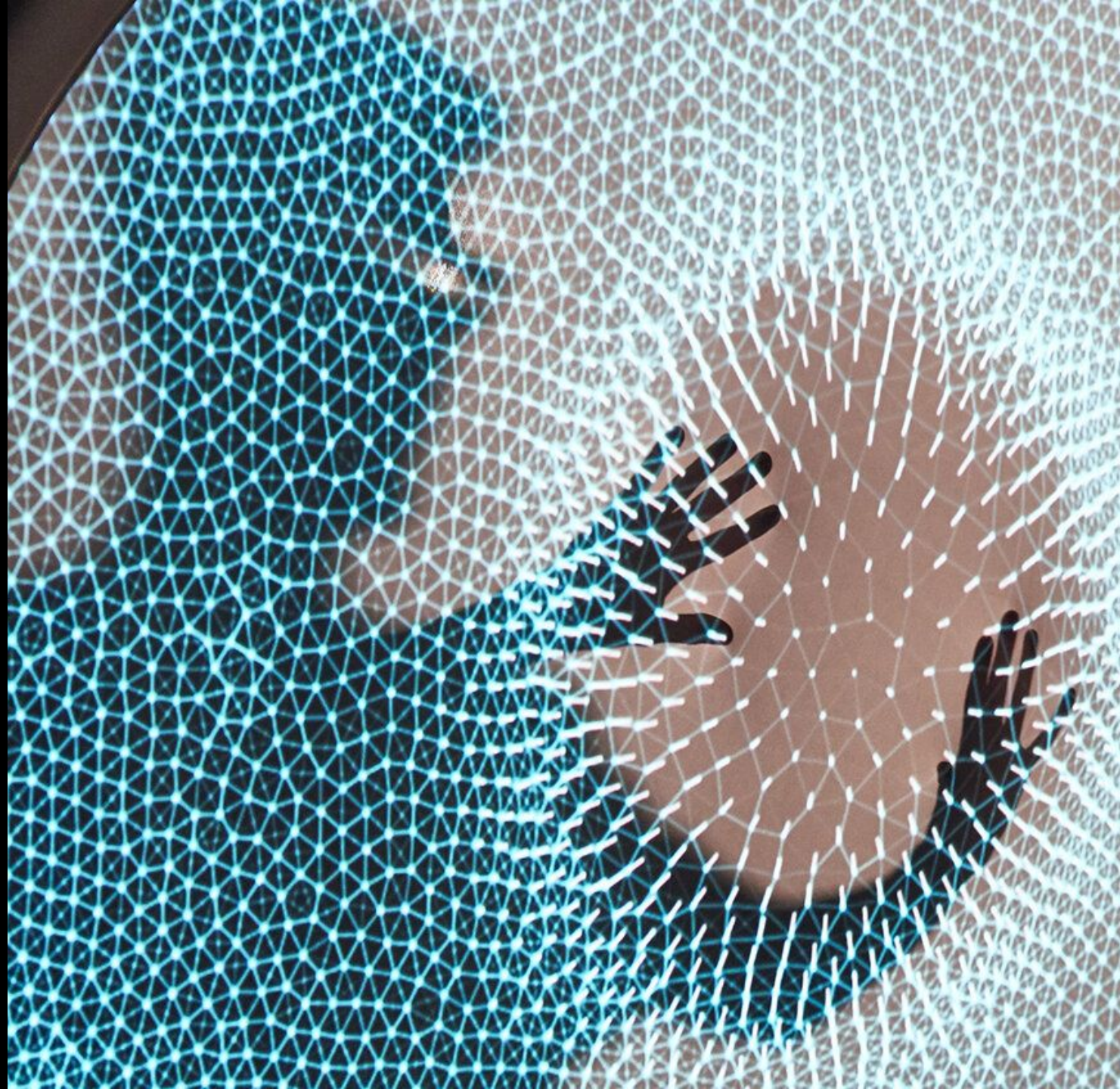


- **Input at 0.5 : Bias value**
- **Low Bias** → Push **values down**
- **High Bias** → Push **values up**

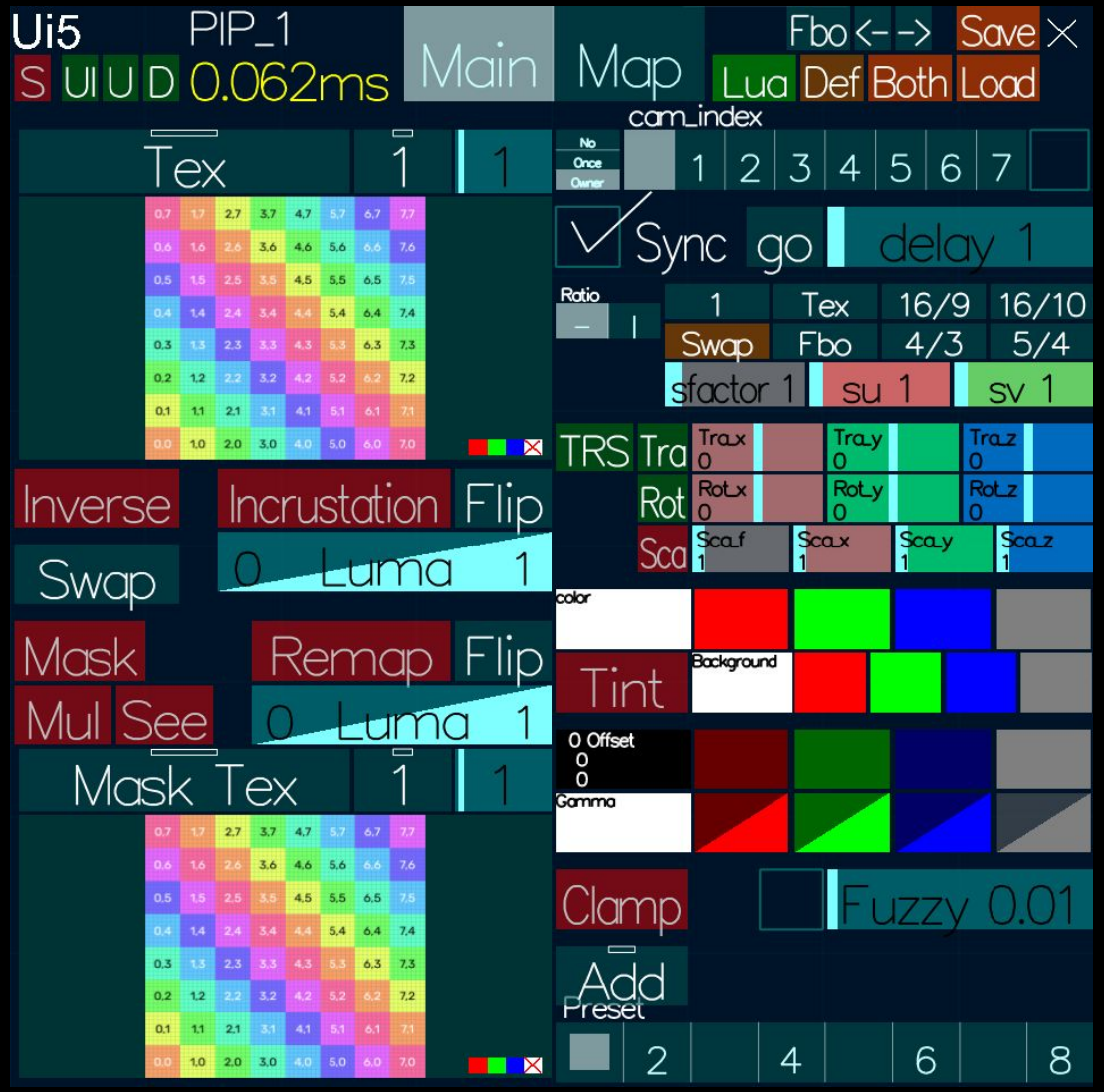
- Double Bias curve (**symetry in the middle**)
- **Low Gain** → Push **value to middle**
- **High Bias** → Push **values to extreme**

AAASeed
An introduction
Part 19
Drawing texture

- MEU PIP
- Ratio and Size
- Translate Rotate Scale
- Blending



MEU PIP (Picture In Picture): Main



- Display a Texture and handle smooth transition
  - Tex : Next Texture
- Luma (Luminescence) Incrustation

- Swap: Exchange texture and Mask
- Mask
  - Remap: like incrustation
    - How Grey map to alpha

  - Flip : Inverse the remap
  - Mul : Multiply luma by the mask
  - See : show only the mask
- Cameras
- Size and Moving setting
- Colors for object, background, offset and Gamma
- Clamp :
- Fuzzy edge / border
- Selector : blending

Ratio and size

- This changes the **sizes used by a MEU to render**

Select direction
the ratio buttons will affect:
here horizontal / u

Ratio buttons
change the slider
su or sv
to match
the ratio asked.
Tex ⟶ Texture

Size Factor:
size in both direction

Size u and v:
size in uv direction

TRS ↔ Translate Rotate Scale

- This controls what is named in 3D: a **transformation**
- **Not moving the camera**
- **Changing the coordonates** of where an **object** is drawn
  - Developer note
    Alter the model (model to world) matrix
- **Combine 3** Types of **transformations**
  - **Tra → Translate**
  - **Rot → Rotate**
    Rotation unit in AAASeed are turns
    0.25 → a quarter turn
    0.5 → a half turn
    1 → a full turn
  - **Sca → Scale**
  - **Experiment and understand in which order Tra Rot Sca are performed**
- Better **keep button TRS Off** if transformation is **not used**

General button
allow global TRS use

Individual buttons
allow each tranformation

MEU PIP : Map

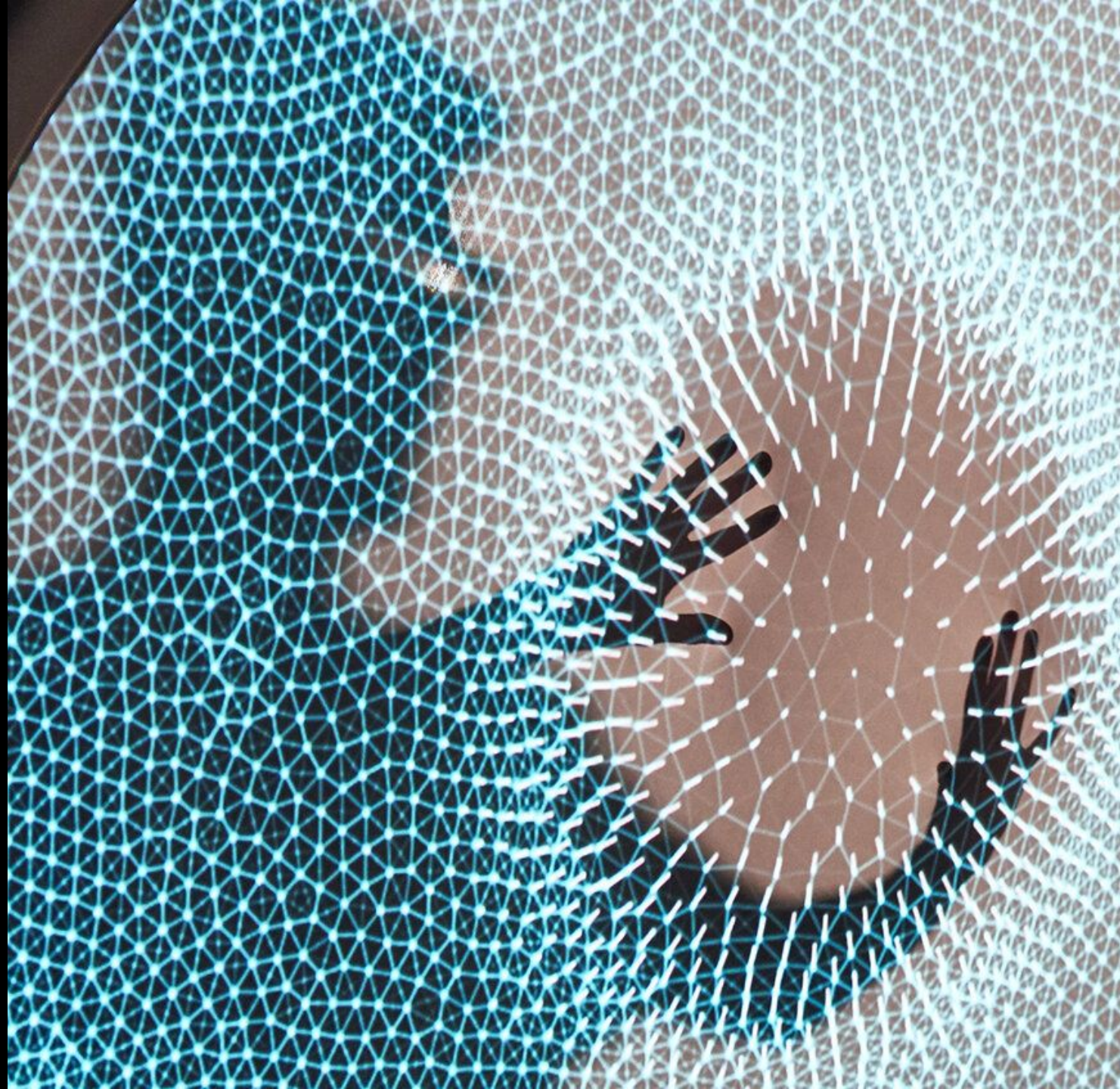
- **nb_u, nb_v : number**
- **U Min Max, V Min Max : stretch the texture**
- **hexa :**
- **top_line :**
- **du, dv :**
- **offsets :**

Blending

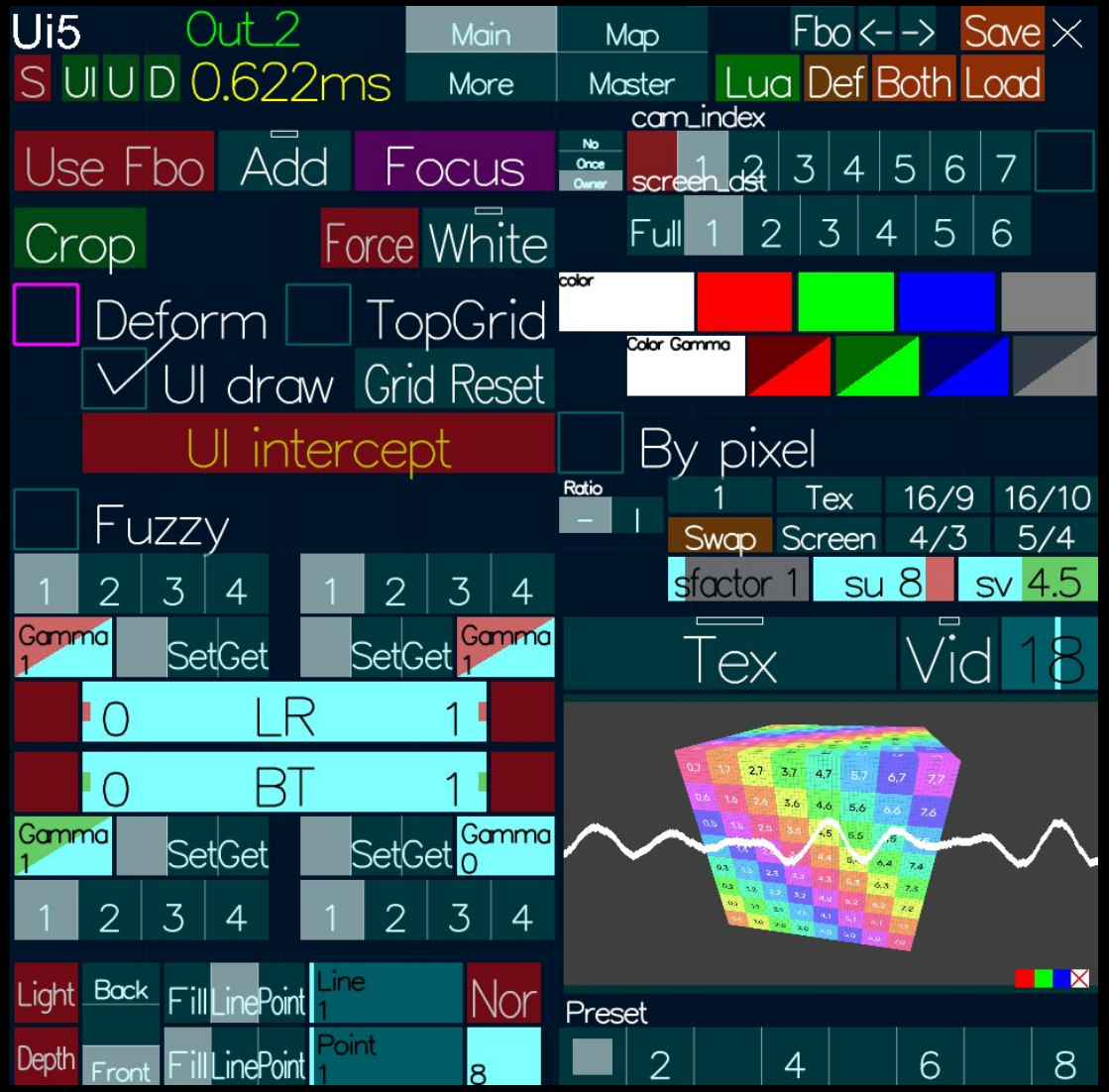
mapping_blend_menu
Min
Max
Add
Sub
RSub
Mul
Screen
Overlay
Darken
Lighten
ColDodge
ColBurn
HardLight
SoftLight
Diff
Exclusion
HSL_Hue
HSL_Sat
HSL_Col
HSL_Lumo
Add

AAASeed
An introduction
Part 20:
Outputting image

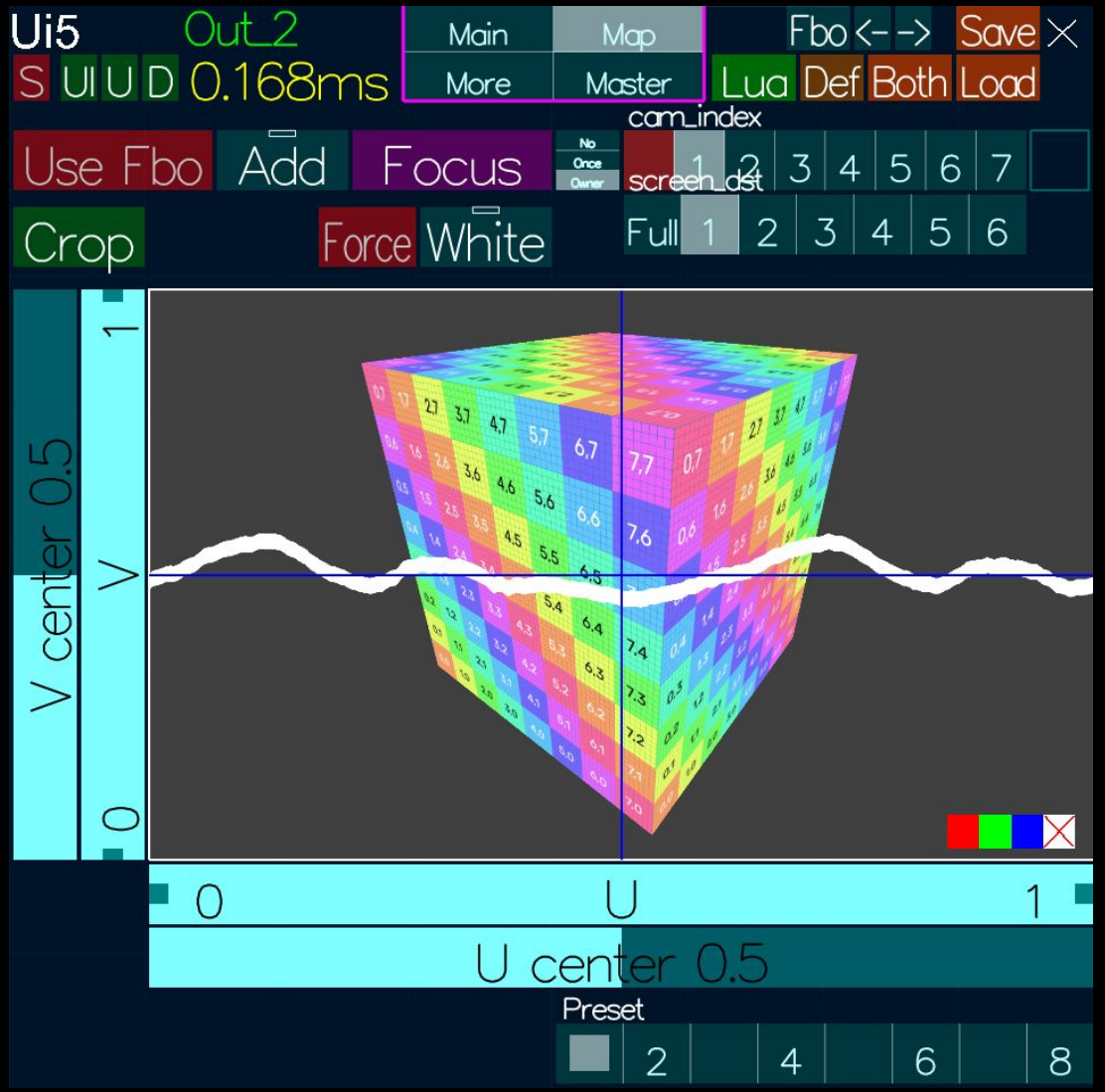
- MEU Out
- rendering



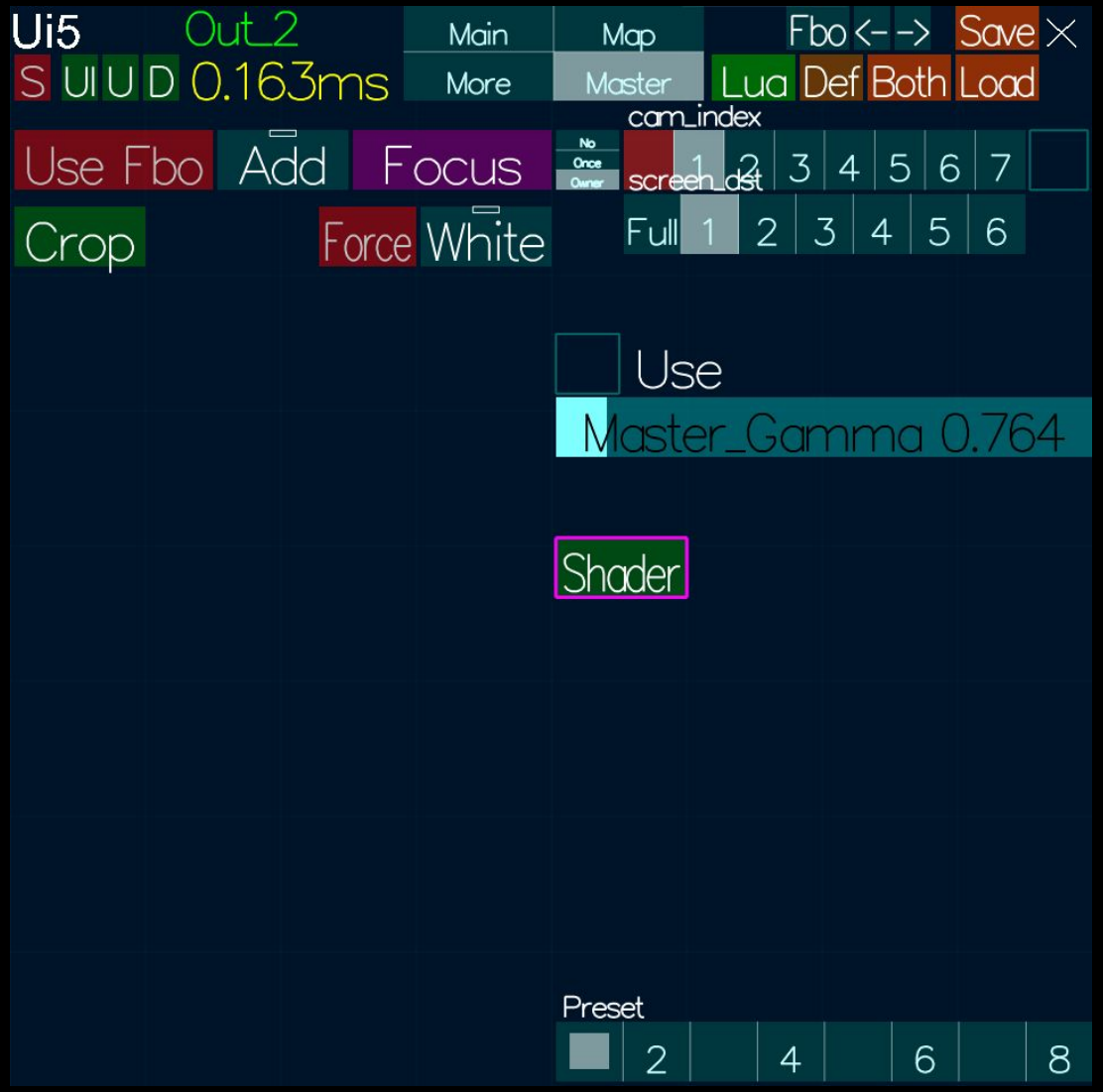
MEU Out: Display an Image in Ouput



MEU Out: Display an Image in Ouput



MEU Out: Display an Image in Ouput

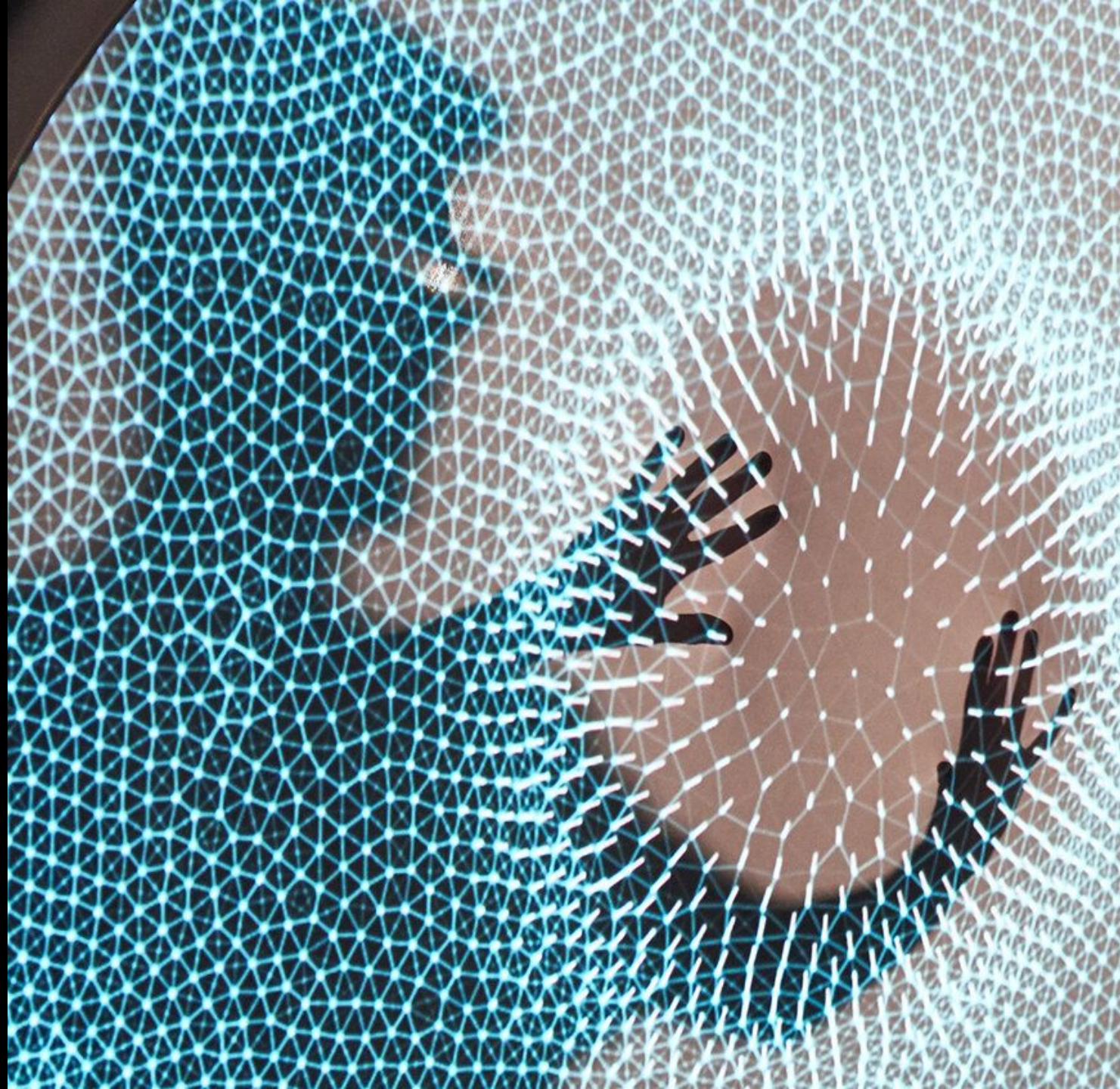


Rendering

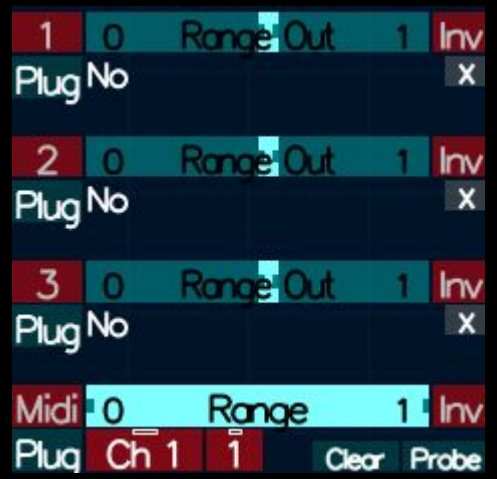
Light
Depth
Back
Front
Fill
Line
Point
Fill
Line
Point
Line
1
Point
1
Nor
1

AAASeed
An introduction
Part 21:
Plugging

- MEU Trax



MEU TRAX

- 1 : open the trax
- Plug : plug it to something
- Control : gain bias threshold
- Visualize input

- Out setting