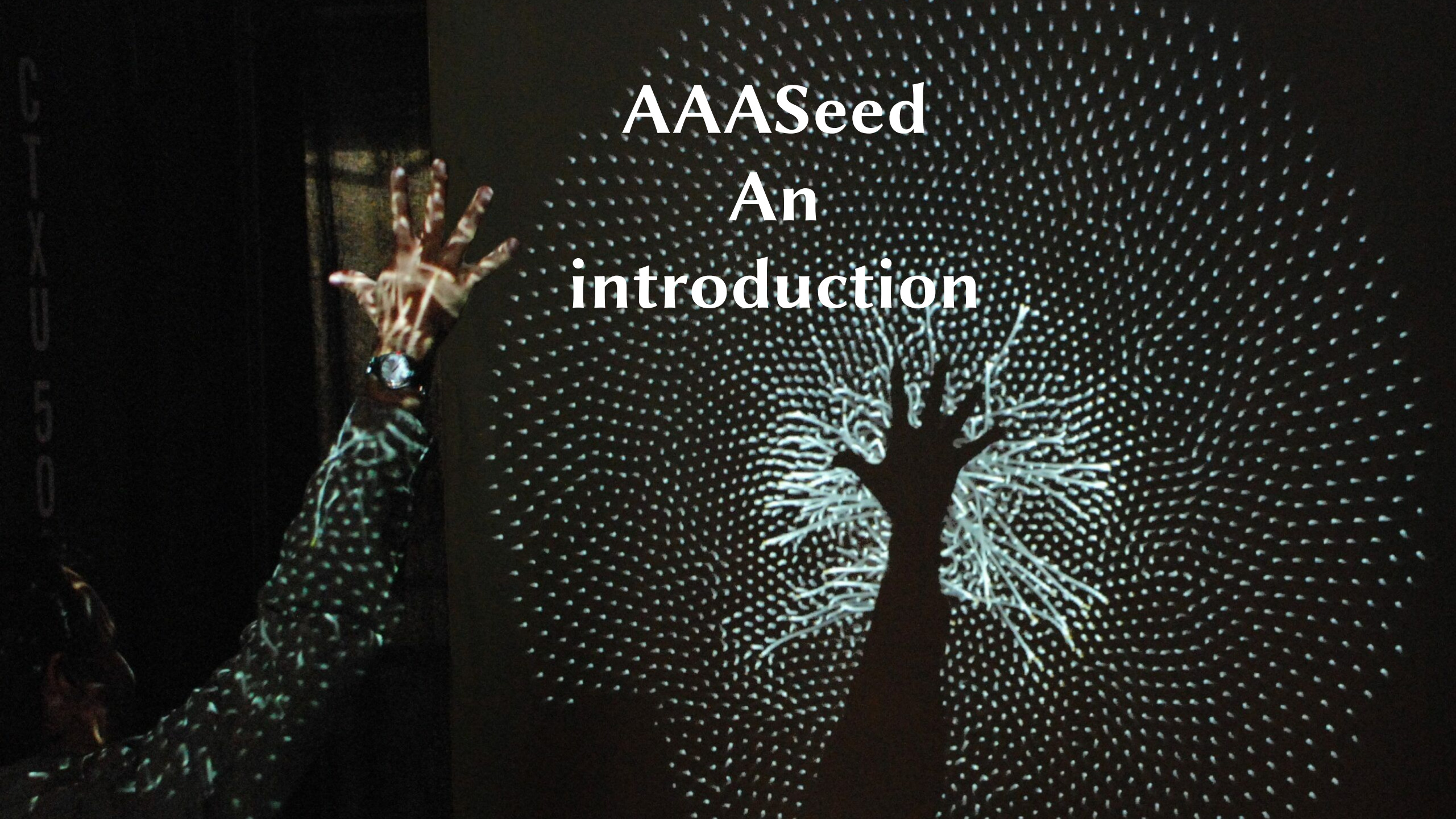


# AAASeed

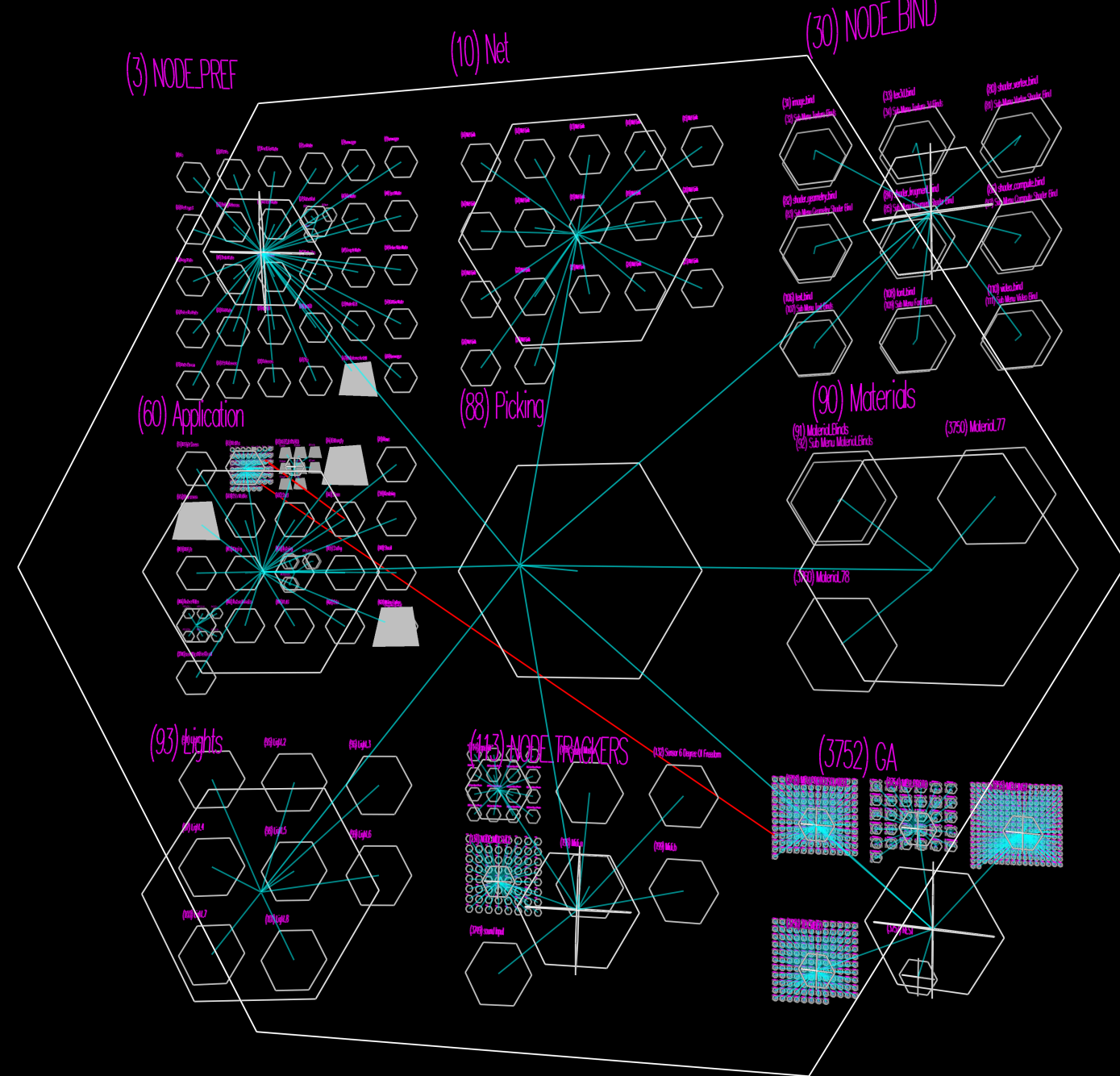
## An introduction





- **AAASeed**  
**An introduction**  
**Part 1:**  
**What is AAASeed**

- **What is AAASeed**
- **Who is AAASeed for ?**
  - **Artist**
  - **CCI**
  - **Developer**
  - **Education, Hobby**
- **User point of view**
- **Shadoks point of view**
  - **Ga Bu Zo Meu**



# What is AAASeed ?

- AAASeed is a powerful, generic toolkit for building real-time processes, creating immersive, high-performance, interactive visual experiences.
- AAASeed is designed for creative professionals and enthusiasts.
- AAASeed is focused on
  - Video Jockeying & Live Mapping
  - Open Data Visualization
  - Interactive Art installations
- AAASeed is the result of 25 years of real-world use and refinement, created by Maa Berriet with significant help from Franz Hildgen.
- It will soon be available as a free, open-source tool under the MIT license.
- Stabilized and improved through an European Project

[ArtCast4d.eu](https://ArtCast4d.eu)

[Early Adopters Program](#)

- AAASeed is build to last

[AAASeed.org](https://AAASeed.org)



# Who is AAASeed for?

## Artist



- Create live, graphic and interactive artworks.
- Build 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**

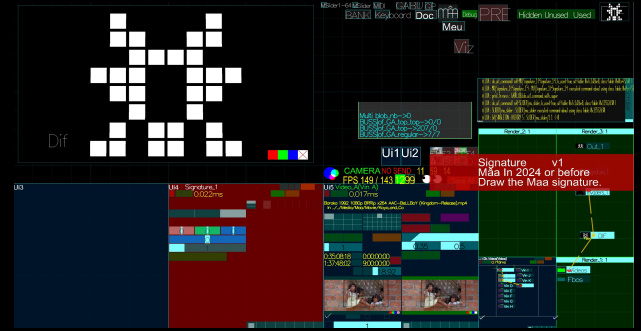
...

- **Free License**  
free digital support for generative/interactive artworks.



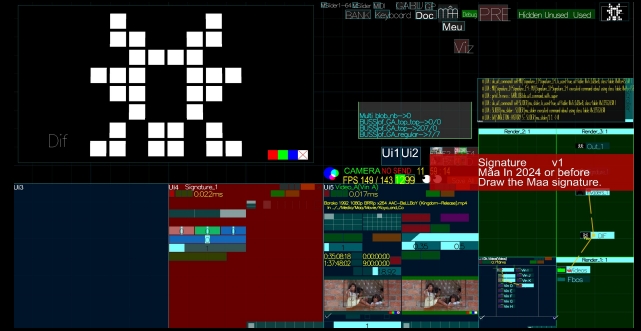
# Who is AAASeed for?

## Creative Cultural Industries



- **Video professionals**  
Control and display videos, images, and text.
- **Immersive environments:**  
Perfect for Multi-screen and multi-machine setups.
- **Typical uses:**
  - Live performance (dance, theater)
  - Broadcast(TV)
  - Digital signage, Interactive screens
  - And more...
- **Seamless Integration: Works live with existing tools, including**
  - Protocols: Spout, OSC, MIDI
  - Software: TouchDesigner, vvvv, Unity, Unreal, Chataigne, OBS studio...
  - Audio: Max/MSP, Ableton Live, PureData/PlugData...
- **Free License**  
Keep your budget focused on production.  
Long-term operational guarantee

# Who is AAASeed for? Developer



- The core of AAASeed is an old school **very robust C++ render graph**.
- **Lua**, the **very fast and flexible scripting** language pilots it and lets you can access the low level functionality:
  - **Window system**
  - **custom C++ and C object**
  - **OpenGL, Glsl Shaders, OpenCl**
  - **Libraries such as OpenCV, Nvidia Flex, dlib, bullet**
  - ...
- A rich **lua virtual machine provide**  
a flexible interface (Ui and code) for editing and customizing real-time processes.
- you can **extend and modify** the system's interface and behavior  
**on the fly**, even while processes are running



# Who is AAASeed for?

## Students, Hobbyists, Teachers



- AAASeed is a rich, fast and complete tool for exploring graphics.
- Learn & Understand:
  - Script and test 3D graphics ideas on the fly.
- Experiment:
  - Dive into live coding, shaders, and advanced mathematics.
- Teach:
  - Use it as a powerful teaching tool for coding and graphics.
- Research:
  - Serves as a great infrastructure for research and development.
- Accessible:
  - Easy to use and more powerful than Processing,  
more graphic primitives coming soon!

# User point of view

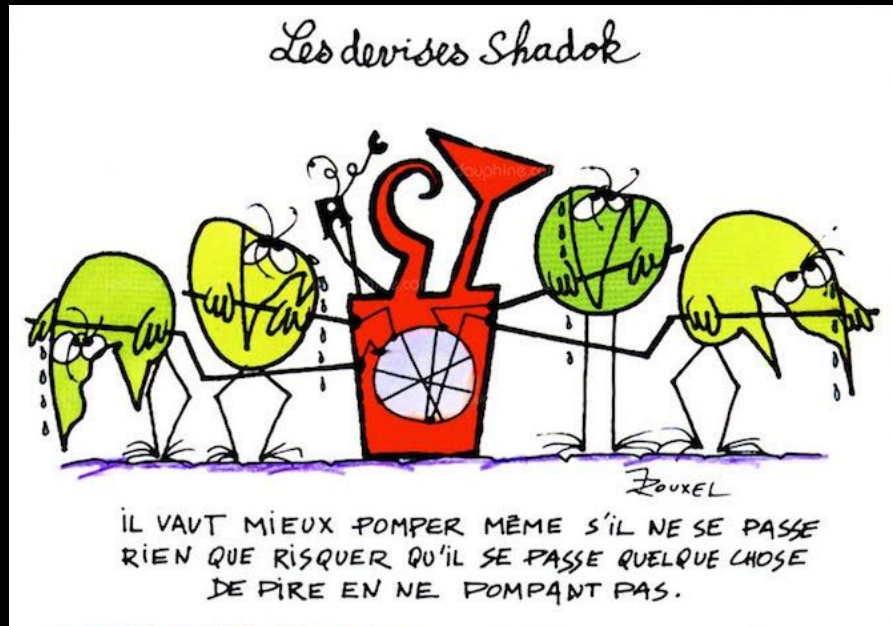
- This a **Windows** application for now  
`AAASeed_Metal.exe`
- Tested against **windows 10 and 11**  
but should also work on Windows 8,7 even XP.
- It can **run on low end machines**  
even an executable for non Avx2 Processor: `AAASeed_Wood.exe`
- But it likes fast machines and **uses the GPU a lot**  
GPU means **Graphic Processor Unit** (the Graphic card)  
**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"



- AAASeed have heritage
  - a playful nod to "Les Shadoks," a classic French animated series.
  - More on this: search Shadoks on YouTube  
[www.youtube.com/watch?v=Sla57Zw-FN4](http://www.youtube.com/watch?v=Sla57Zw-FN4)
  - Thanks to [aaa](http://www.aaaproductio.fr)production ([www.aaaproductio.fr](http://www.aaaproductio.fr))
- You are about to enter another world: welcome
  - if it can look weird at the beginning
  - It will make sense
    - it have its own logic
    - It have being tested in the real world
      - by very different users
      - in a wide range of conditions

# Shadoks point of view

## Ga Bu Zo Meu

- **New world, New vocabulary**
  - Avoid confusions with existing concepts**
  - Simple and Compact (shorter code and sentence)**
- **GA: Global Action**
  - is the **top level** of the system
- **BU: Box User**
  - are the basic **element of the Ui (User interface)**
    - button, slider, text...
  - BUs are regrouped in BUS** (a group of BU)
  - a BU can contain a BUS**
    - this way a BU can contain other BUs to compose a more complex Ui
- **MEU: Module Editable Unit**
  - functional editable blocks**
    - the **AAASeed atomic blocks** to build your processes
  - can be represented in their short form the **MU: Module Unit**
  - BUS and BU are used to define the MEU Ui**
- **ZO: We lost the ZO**
  - «On a perdu les ZOs»: but we will find it, promised



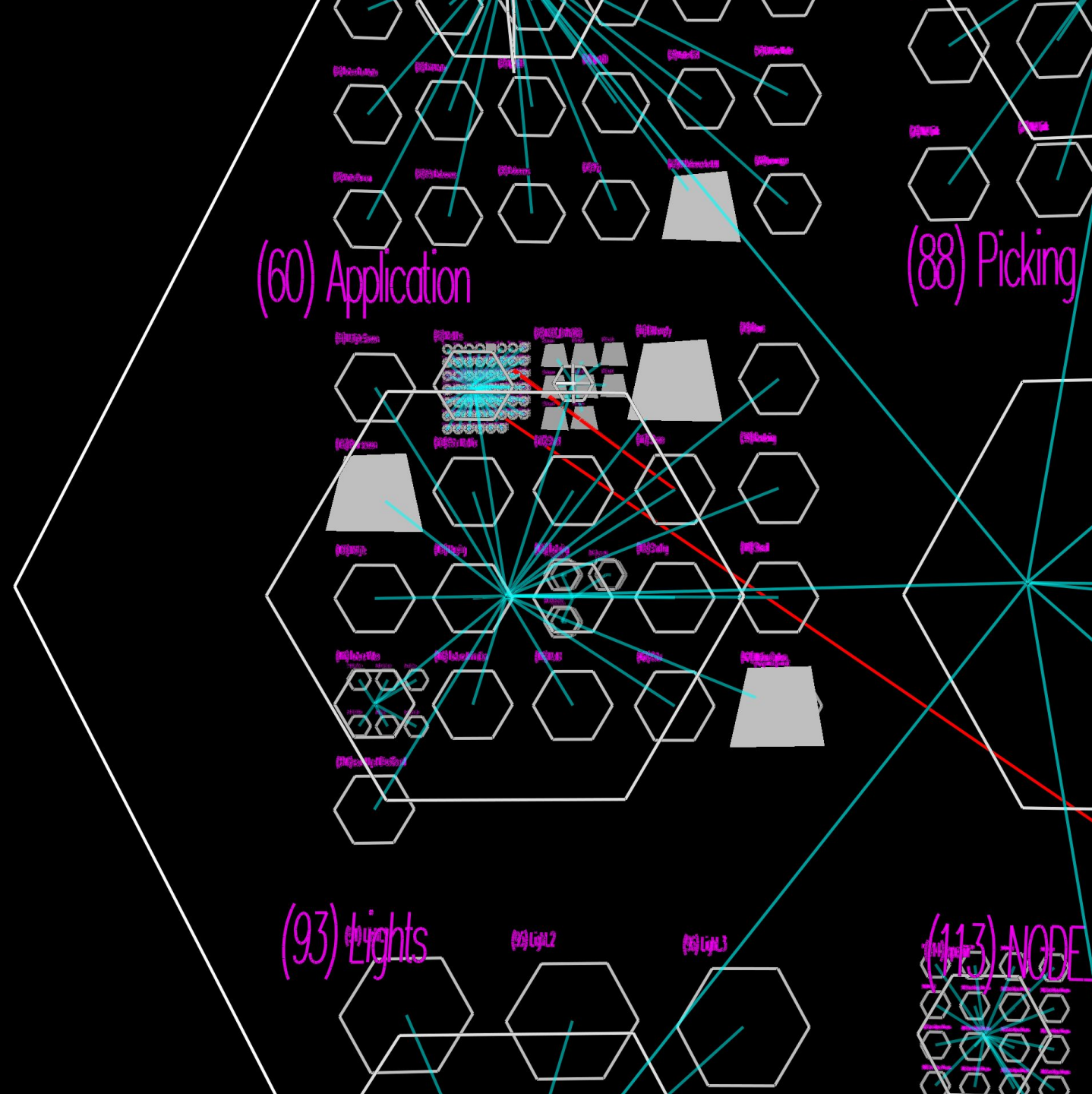


# AAASeed

## An introduction

### Part 2: Installation

- Installation
  - **OS and machine**
  - **Required software**
  - **Other software**
  - **AAASeed / File structure**
  - **Multiscreen**
- First Launch & Basic Controls
- Trouble shooting
- AAASeed Folder
- AAASeed Sources



# Installation: Os and machine

- This a **Windows** 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**  
GPU means **Graphic Processor Unit** (the Graphic card)  
**Loves NVidia** but **functions on integrated Intel**  
**supports Amd** most of the time (getting better on Amd every month)





# Installation: Required software

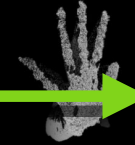
- To run AAASeed, you need to install a few essential applications first:
  - **Klite Codecs**  
**Decompresses video files.** AAASeed uses it for video playback (through DirectShow)  
[www.codecguide.com/download\\_kl.htm](http://www.codecguide.com/download_kl.htm)  
we use klite Mega, but other options should function too  
Use the default installation options.
  - **7z**  
**Compresses and decompresses .7z files,** which AAASeed uses for updates  
[www.7-zip.org](http://www.7-zip.org)  
Get the 64-bit x64 version  
Run the downloaded executable.
  - **Visual Studio Code**  
**This is the default text and code editor for AAASeed**  
[code.visualstudio.com](http://code.visualstudio.com)  
Get the Windows version  
Run the installer  
Install extensions for Lua, GLSL, and OpenCL syntax highlighting  
For best results, open your entire AAASeed folder as a workspace to easily search and find files  
open AAASeed\_VSCode.code-workspace is one way to do this  
Notepad++ and Sublime Text 2/3 were used and should still function  
configure in Flatland at Pref/MASTER/Lua/lua\_master/editor

# Installation: Other software

- The following tools are **not required to run AAASeed** but are **highly recommended** for specific workflows:
  - **PureData and/or PlugData: Audio software**  
Used for **running patches related to MIDI to OSC** (conversion and sound analysis).  
Needed to run patches in AAADoc/PureData:  
Midi to OSC  
Sound analysis to OSC  
[puredata.info](http://puredata.info) and/or [plugdata.org](http://plugdata.org)
  - **XnView: An image browser with powerful batch processing and conversion capabilities.**  
[www.xnview.com/en/](http://www.xnview.com/en/)  
MP or Classic
  - **Wings3d: A 3D modeler useful for exporting .obj files**, derived from Symbolics/Nendo.  
[www.wings3d.com](http://www.wings3d.com)
  - **Blender: The industry-standard open-source 3D editor.**  
[www.blender.org](http://www.blender.org)
  - **OBS Studio: Free and open source software for video recording and live streaming.**  
[obsproject.com](http://obsproject.com)
  - **Spout: Ultra-fast, realtime video routing for Windows**  
[spout.zeal.co](http://spout.zeal.co)
  - **Spout To NDI: a set of programs that allow Spout senders and receivers to share video over a network**  
[leadedge.github.io](http://leadedge.github.io)

# Installation: AAASeed

AAASeed



- Download the latest files from [AAASeed.org/files](https://AAASeed.org/files)

Run `AAASeed_Setup.1.3.1.exe` (or the latest version)

**A system reboot is required after the first installation.**

**Download and add `libcurl.dll` and `cpr.dll` to `AAAD11` Fo**

Update: There is no automatic updater yet.

**To update, download the latest `AAASeed.7z` file**

**then replace your existing `AAASeed` folder with the new one**

**You can rename previous or current `AAASeed` folders as you w  
keep different versions if you want**

- Understanding the general file structure

-AAAFoundation	<b>Stores all AAASeed related content and your personal work.</b>
-AAAD11	<b>Contains external software libraries (DLLs). The installer adds a system path to this folder,</b>
-AAASeed	This folder contains the core executables and code maintained by the AAASeed team. <b>Do not save your personal work here.</b>
-AAASUser	<b>Your user preferences are stored here.</b>
-APPs_Guest	This is where you should save your own applications (APPs).
-Media	<b>Place all of your content (videos, images, sounds, etc.) here. Eventually use sub folder by projects</b>
-Install	<b>Associated installers, drivers...</b>

- It is possible to install “manually” (no installer) from a local drive or USB Key  
Copy the folders and add the DLLs path to the User or System path

AAASEED PRODUCTIONS VIDEOS MÅA'S CORNER FILES NEWS CONTACT

## FILES

### AAASeed Downloads

- [AAASeed\\_Setup.1.3.1.exe](#)  
current installer 2025 April 8th
- [AAASeed.7z](#)  
version 2025 April 30th 5pm
- [AAASeed\\_Introduction.pdf](#)  
version 2025 April 30th 5pm
- [Main\\_x64\\_2025\\_April.7z](#)  
separate folder with the DLLs from  
`AAASeed_Setup.1.3.1.exe`
- [libcurl\\_and\\_cpr.7z](#)  
2 DLLs added since the installer `AAASeed_Setup.1.3.1.exe`



# Installation: Multiscreen

AAASeed supports up to 6 screens

On desktop

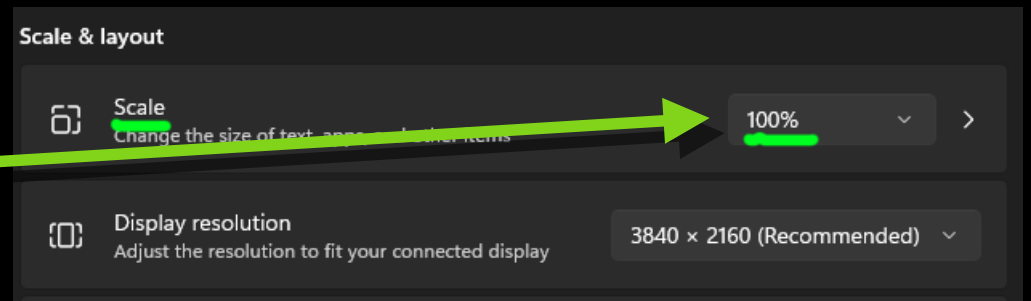
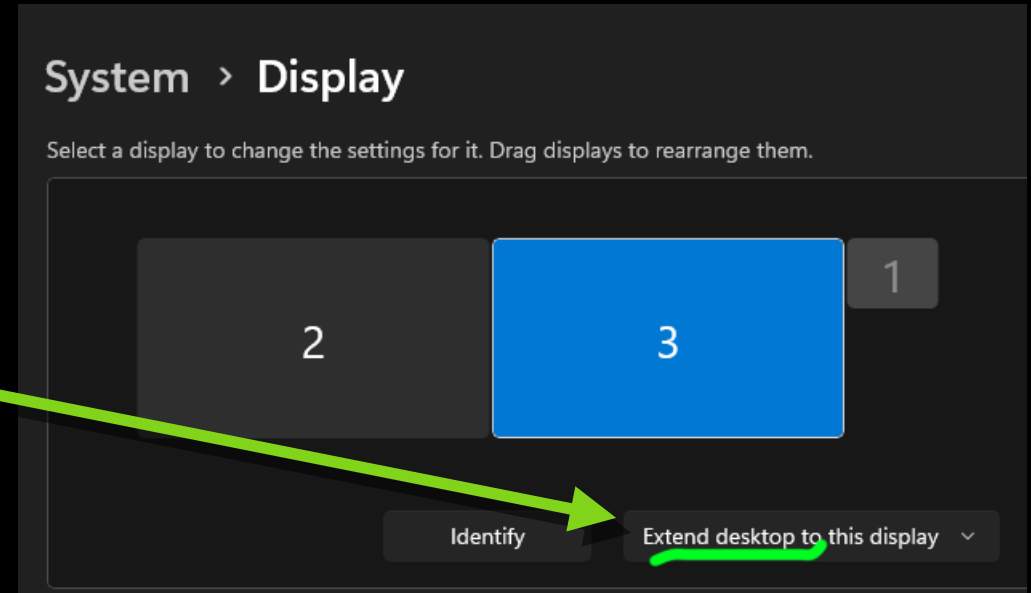
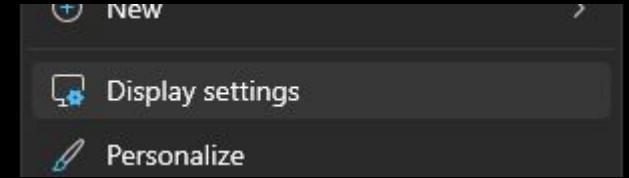
right click menu

Display Settings item

Dialog

- Choose **Extend desktop** mode
- Be careful with the **screen alignment**  
**Horizontal and Vertical**
- Choose the **right resolution**

**Note: Was mandatory before 2025 May**  
**Scale all screens to 100%**  
**seems ok now, testing**



# First Launch & Basic Controls

- Run

Launch `AAASeed_Metal.exe`

For machines without AVX2 extensions, use `AAASeed_Wood.exe`.

`AAASeed_Metal.exe` will crash with no message on these machines.

- Switching display modes:

Press `w` → switches between windowed and full-screen mode.

Press `Tab` → toggles the Flatland interface on and off.

Press `Ctrl E` → enters edit mode  
indicated by a green lock at the top-left of Flatland.

- Exiting:

Press `Esc` twice fast (Double Escape)  
→ quits and save your global settings.

Press `Alt F4` or right-click the mouse and select Exit  
→ quits without saving.

# Trouble shooting (Archive)

- Missing DLLs:

If you see a missing DLL error (other than a VC...dll file) you may need to add the AAAD11 folder to your system's environment variables:

Copy the Dll path from the folder where they are

Go to PC / Properties / Advanced system settings

Choose Environment Variables

Choose User variable/Path (User or System) / Edit...

Choose New

Paste path and eventually change order

Validate : Choose Ok then Ok then Ok

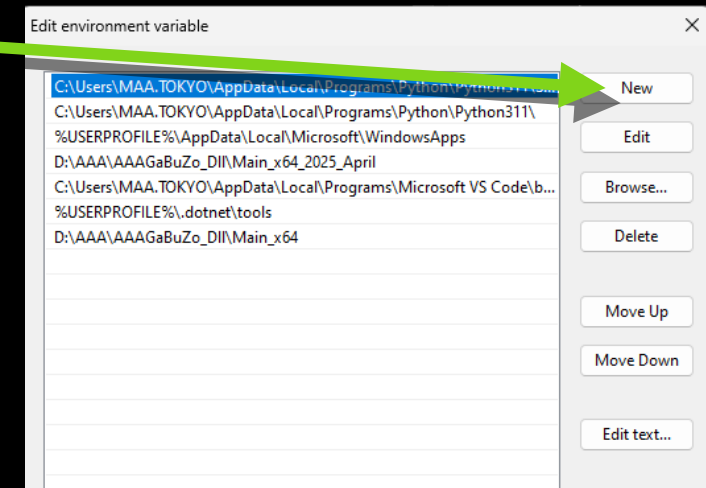
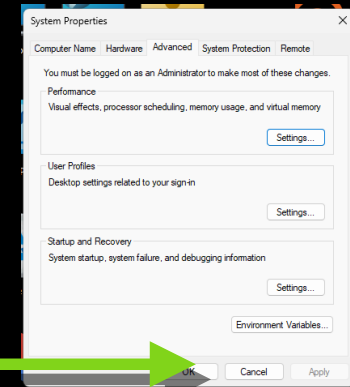
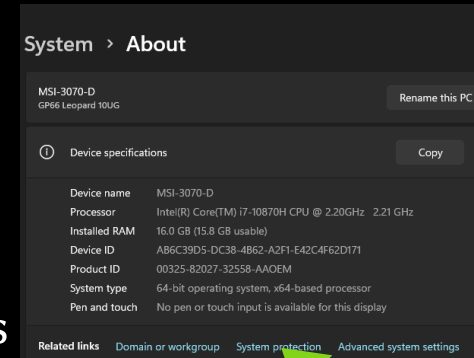
Launch AAASeed\_Metal.exe

- Missing VC ... .dll

Run the appropriate VC\_redist\_2015\_2022...exe installer.

- Missing VCOMP120.dll

Run vc\_redist\_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`
  - Details the installation process and tree structure
  - Document the start sequence
  - Preview in VSCode
- `ReleaseNotes.md`
  - Documents the the changes and new features from one version to the next.
- `Version.txt`
  - A simple text file that shows the current version number of the installer.

# AAASeed Folders: Folders inside

- AAAAPPs Folder
  - **APP** is like a document or project in AAASeed.  
An App ↔ A **folder** with a file `default.layerss_param` inside  
Examples, Tutorials...  
**No way to save an APP for the moment: DUPLICATE FOLDER !!!**
  - **Regroups APPS maintained by the core team**
- AAADoc Folder
  - stores the documentation. Far from finished (state in 2025 May)
  - **Lua API**  
`lua_aaaseed_draw.lua`  
`lua_aaaseed_interface.lua`
- AAASKernel Folder
  - **this Developer space contains core resources for AAASeed**  
lua code, shaders, OpenCL, Fonts, Textures...
  - in lua `aaa.dir.get_dir_kernel()`

# AAASeed Folders: going Up

- **AAASeed** the Startup folder (where the executable starts)

in lua `aaa.dir.get_dir_start()`

- **AAAUser** folder

- **AAASeedWho.txt** made of **3 lines**:

UserName

Where **global stuff** are saved (pref, master, binds...)

in lua `aaa.dir.get_dir_user()`

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

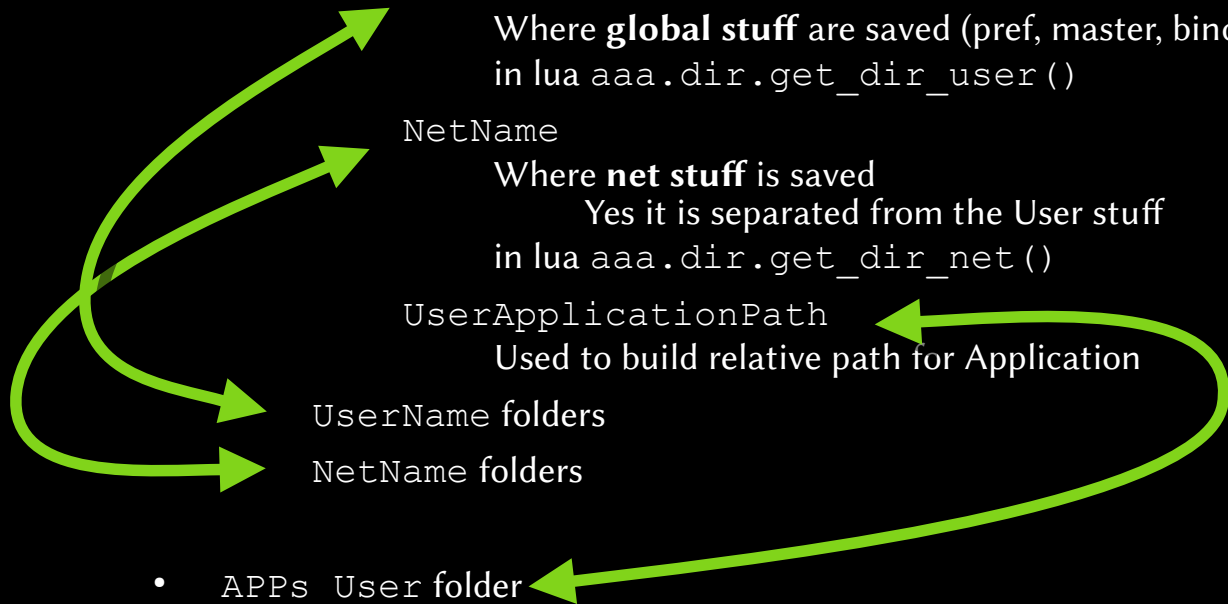
- **APPs\_Guest** by default

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

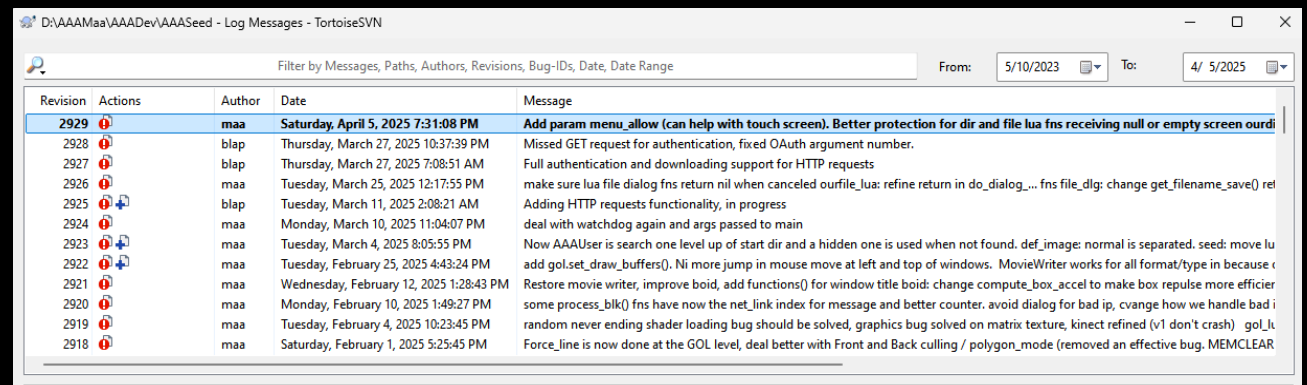
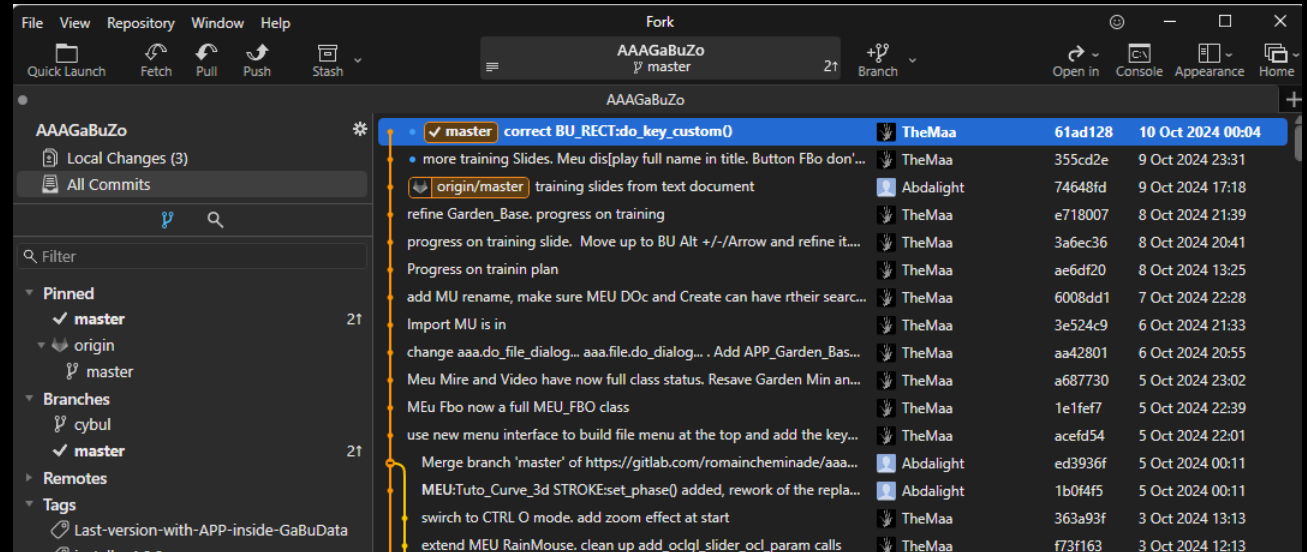
## Svn

-AAASeed.exe and dll

-C++

-will move to a git

request Mâa if you want it



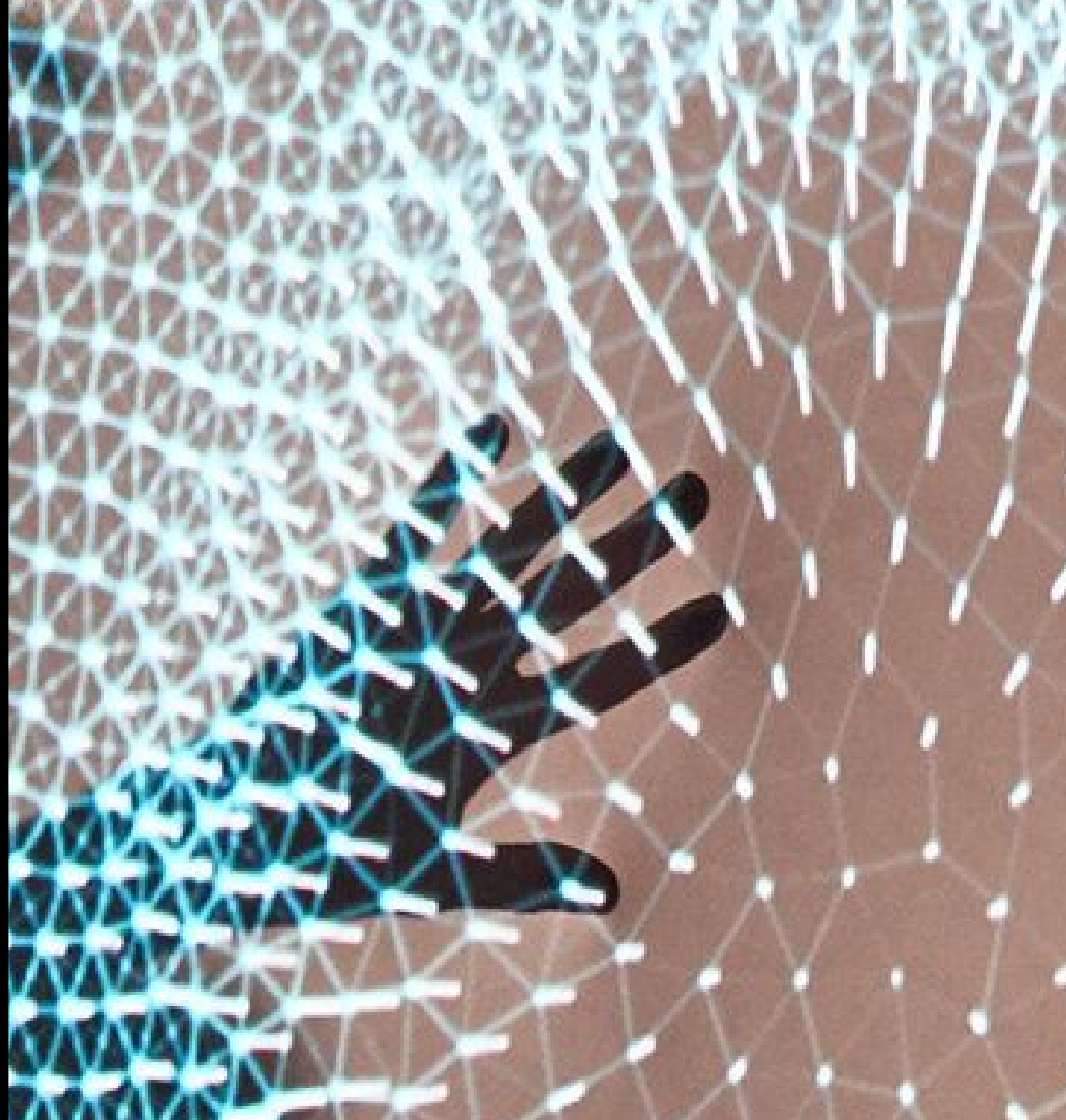


# AAASeed

## An introduction

### Part 3: Baby steps

- **Update AAASeed**
- **First steps**
- **Flatland 101**
  - Example
- **Multi screen**
  - Flatland
- **Lua Error Dialog**
- **APPlications**



# Update AAASeed

- Get AAASeed.7z

**AAASeed.org**

FILES section

Certain browser like Chrome (2025 July) block the download: see your browser security documentation

- Replace existing AAASeed Folder

Reminder

**you can rename AAASeed folders**

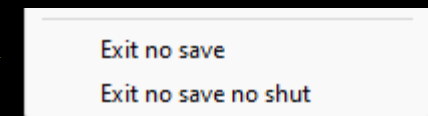
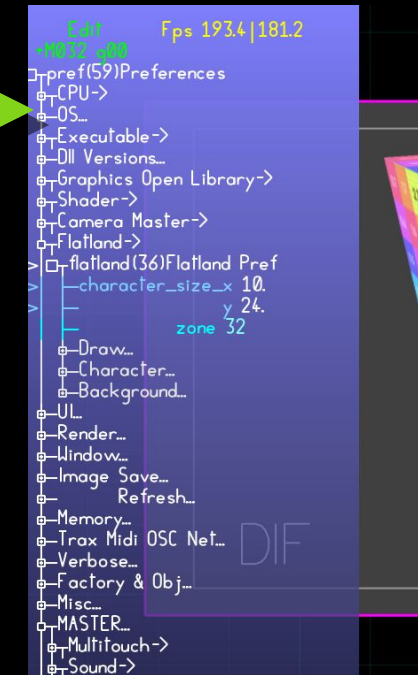
**current version (July 2025) needs 2 more dlls**

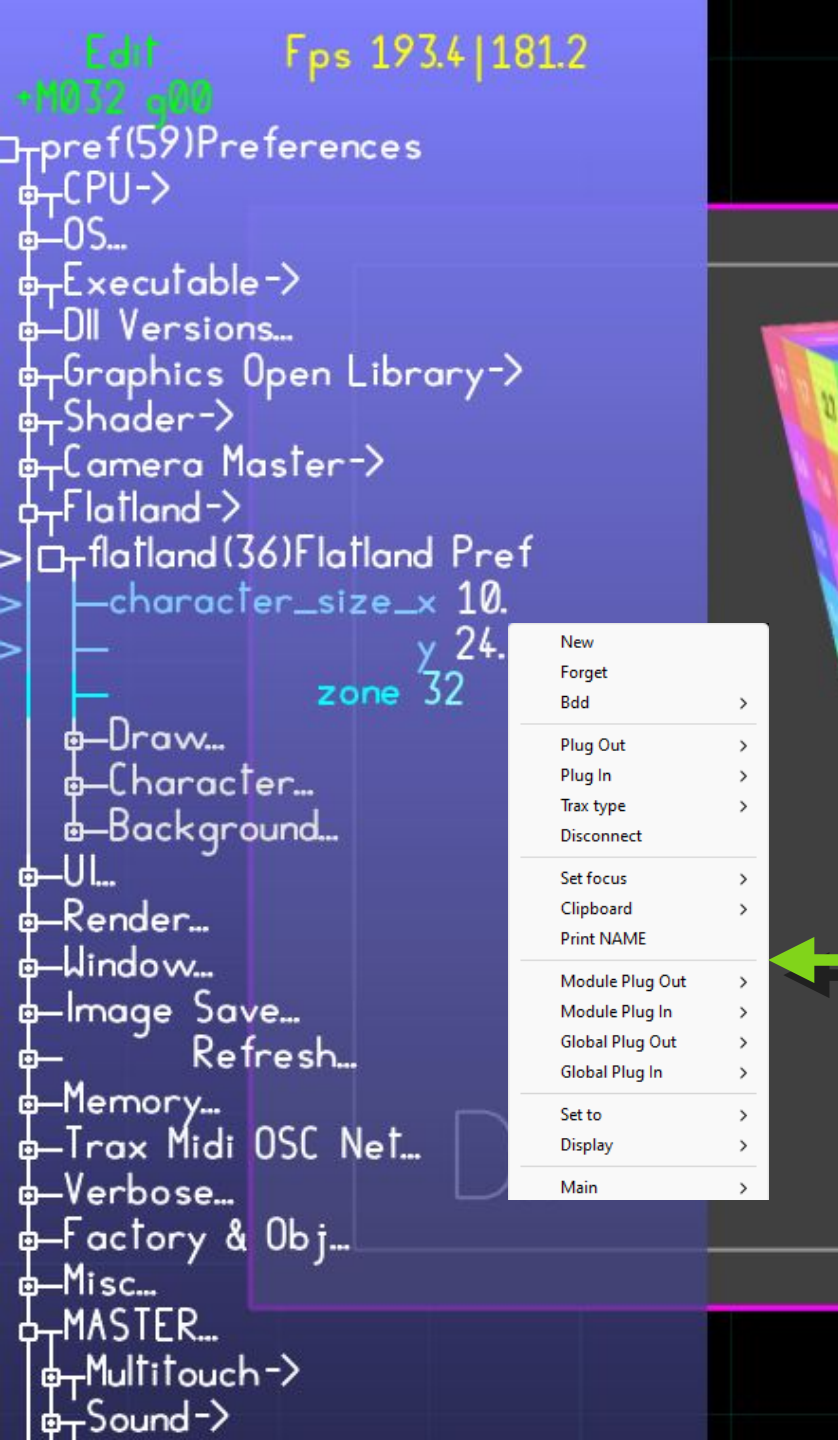
libcurl.dll and cpr.dll

- Available also at AAASeed.org FILES section
- Put it in AAAFoundation/AAADll

# First steps

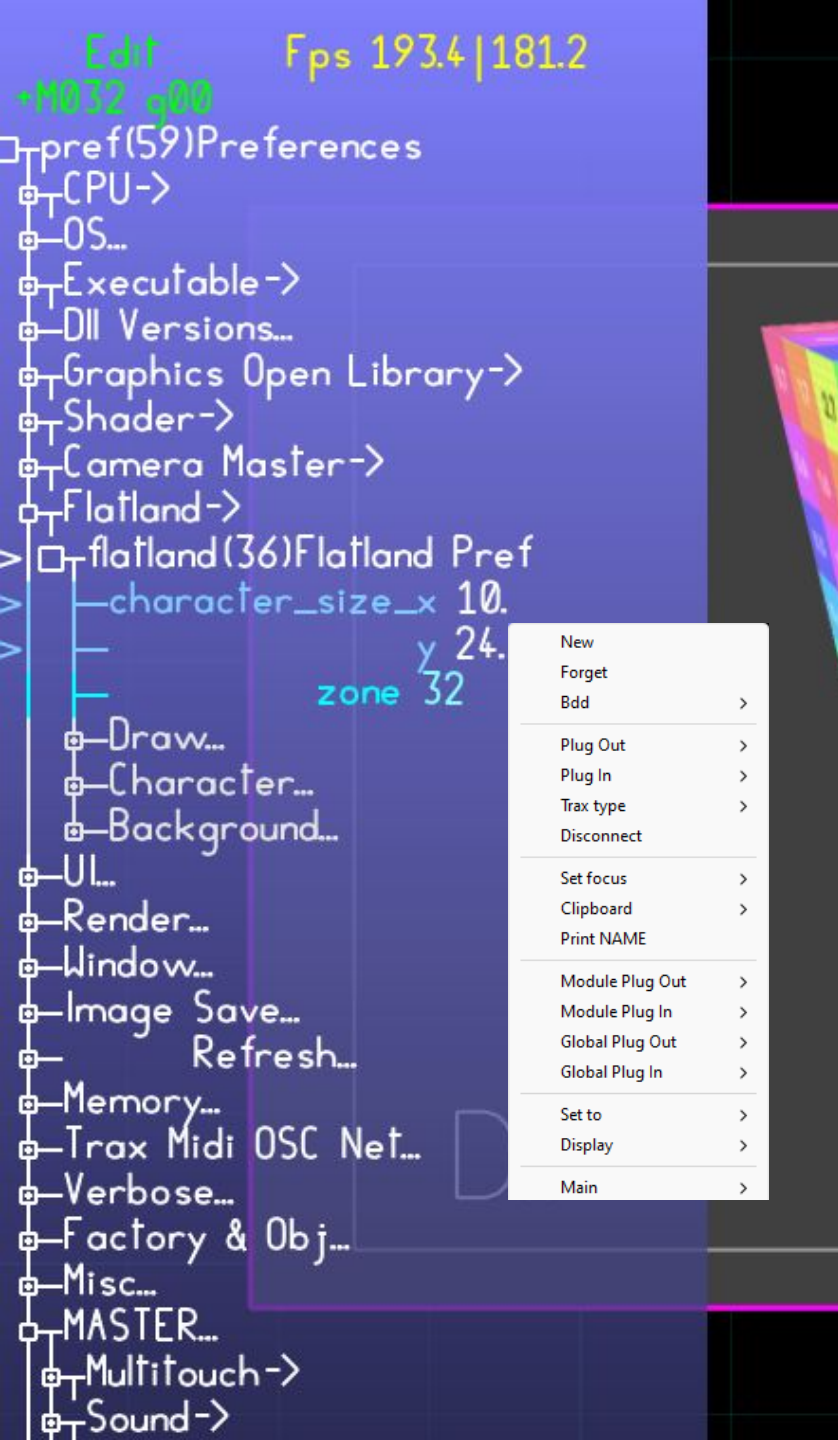
- keyboard first step
  - w like (w)indow →  
switch between window and full screen Module
  - Tab → switch Flatland On and Off  
Ctrl e → like (e)dit flip edit mode (more later on this)  
see feedback at the top left of Flatland  
**Red LOCK**    **Green Edit**  
leave it as **Edit**
- Quit, Exit ...
  - 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 101

- **Original AAASeed Ui** from previous century
  - Name** from « **Snow Crash** » book by Neil Stephenson where the **C++** can be seen and used
  - made of **c\_obj\_ui** and **param** (for parameter)
  - Tree Made of params
    - developer will say render graph
    - a whole world we will explore later
- example: change values in Flatland (next slide)
- **Keys**
  - **Tab** → **switch Flatland On and Off**
  - **F10** → **focus flatland on Preferences**
  - **Ctrl F10** → **focus flatland on Preferences at start**
- **Right Mouse Button** → **Menu**
  - in Flatland** → **Param menu**
  - out Flatland** → **Main menu**
  - GaBuZoMeu menu** ← **Beginning of Main menu**
- **Flatland** have precedence for events and keyboard
- Needs to be off to use keys with **GaBuZoMeu**



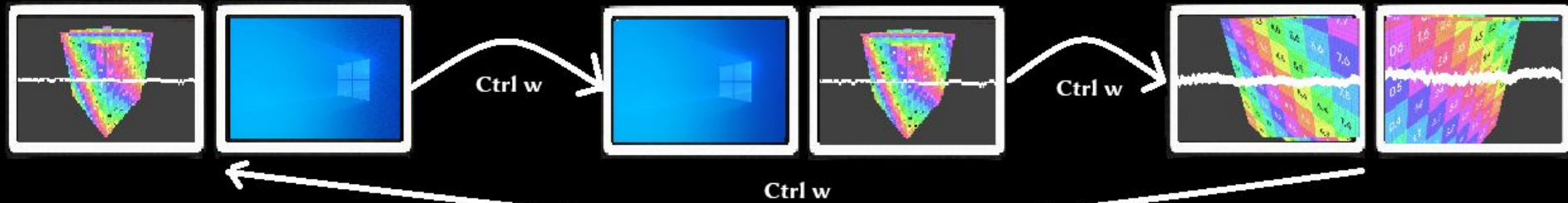
# Flatland example

- change values `character_size_x/y`
  - F10 → Preferences
  - Wheel → Scroll
  - Click (left Mouse Button) on Flatland → select / open
  - Click on `character_size_x` maintaining the mouse button down then turn around the starting point to change the value release the mouse button
  - Double Click → open an edit dialog
  - Keys + - \* / → change the value
  - Ctrl z → undo (only last change 2025 July)
- Reminder
  - Flatland have precedence for events and keyboard
  - Needs to be off to use keys with GaBuZoMeu
  - So leave it off for now → Tab

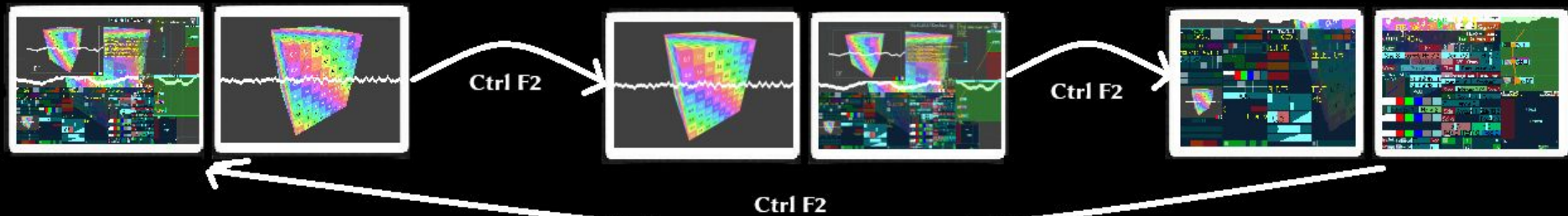


# AAASeed and Multi screen

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



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



- Same principle applies with more screens (up to 6)

# Multi screen and Flatland

- Flatland by default on left of AASeed Window which can traverse several screens
- If it is a problem

switch AASeed window position

See previous slide

or change Flatland position

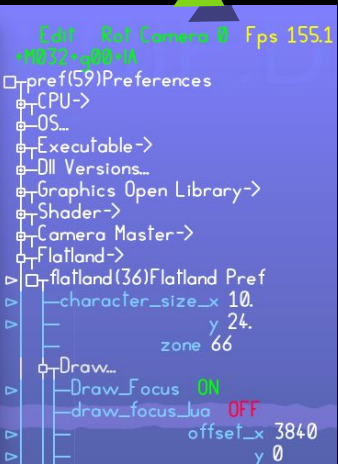
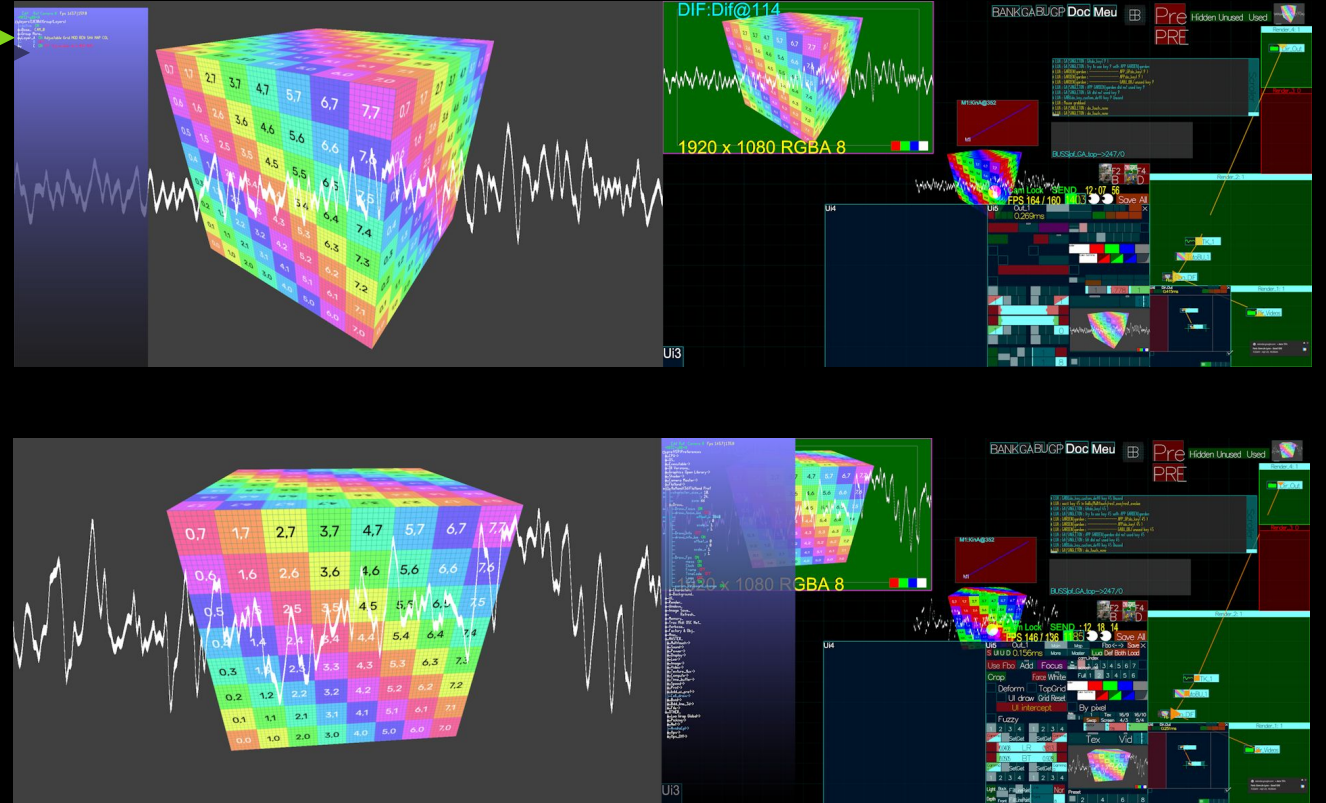
Flatland (Tab)

Preferences (F10)

Flatland/Draw

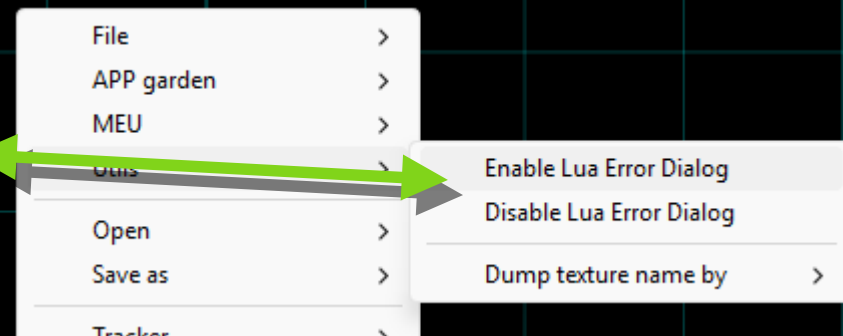
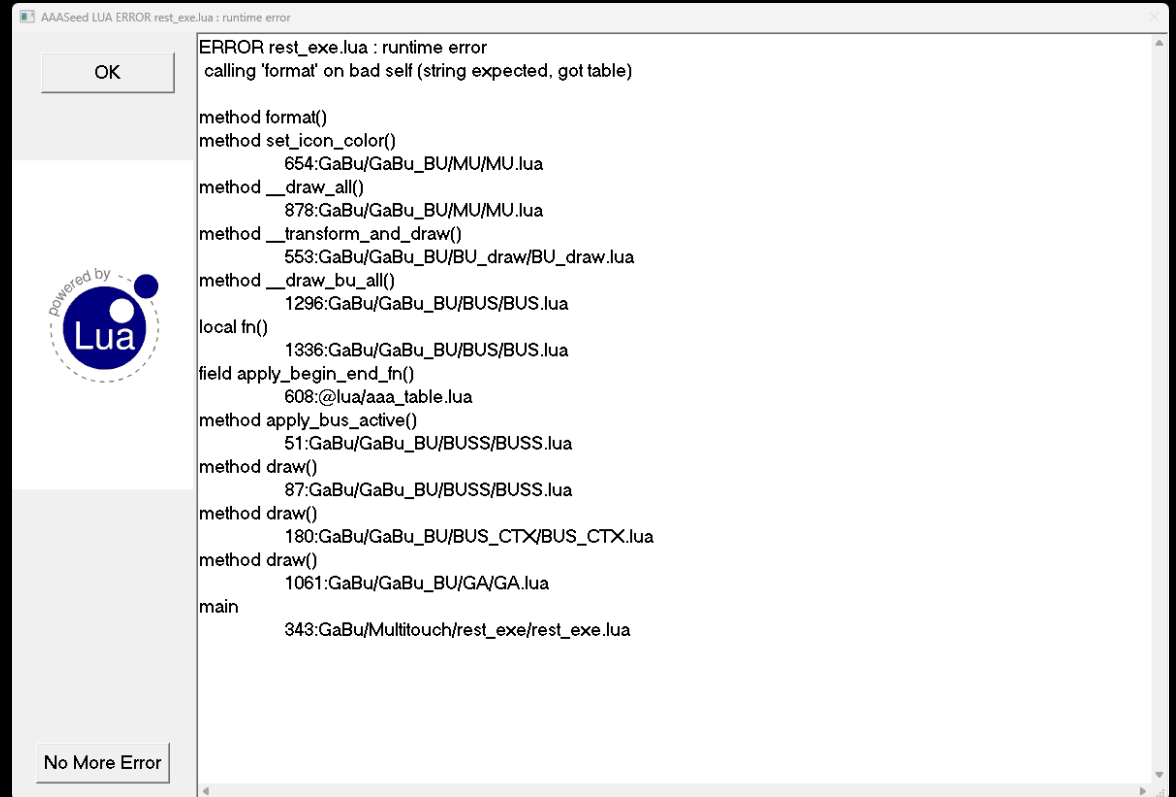
draw\_focus\_offset\_x/y

Careful it can make Flatland invisible  
(drawn outside)



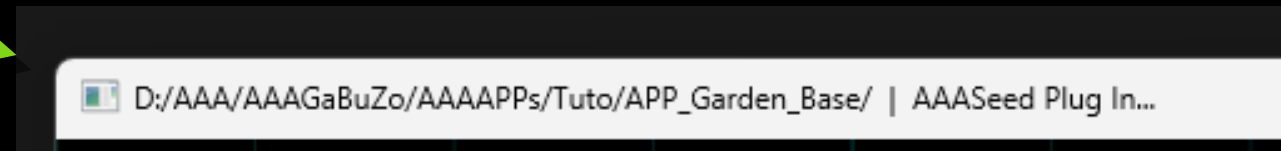
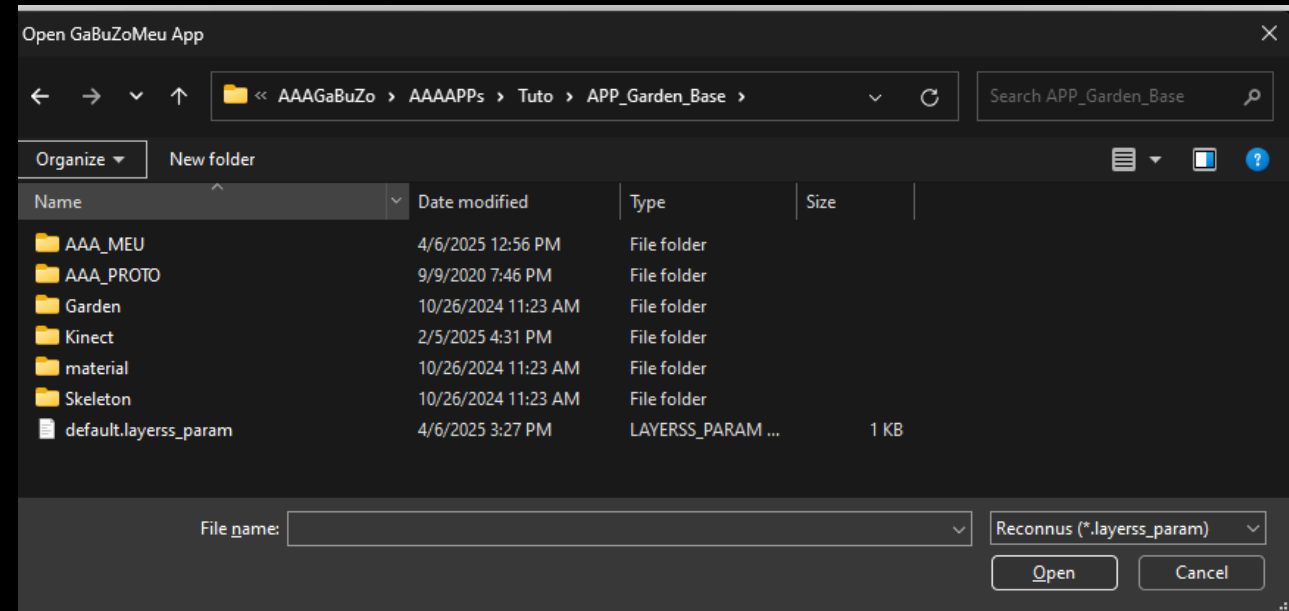
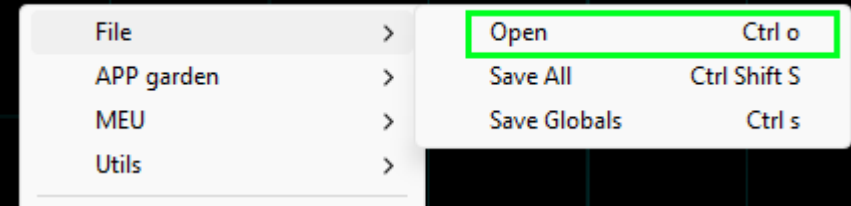
# Lua Error Dialog

- **Lua scripts run GaBuZoMeu**  
Errors can trigger an error Dialog
- **Dialog**
  - Display the error and its call stack, open associated editor at the error location. We use VSCode by default.
  - Correct the error in the text editor if you understand what you do. Save the modified file.
  - Click Ok  
The eventually modified version will be used.  
Future errors will triggers this dialog.
  - Or Click No More Error  
Errors will not triggers this dialog anymore  
not recommended if you don't know what you are doing.  
All future errors will be displayed in the terminal window. Most likely AAASeed will start to behave strangely and you will not understand what is going on.
- **Menu Utils control same lua Error Dialog preferences**
  - Always Enable to see problems
  - Disable for permanent installation



# APPlications or APP

- Open 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



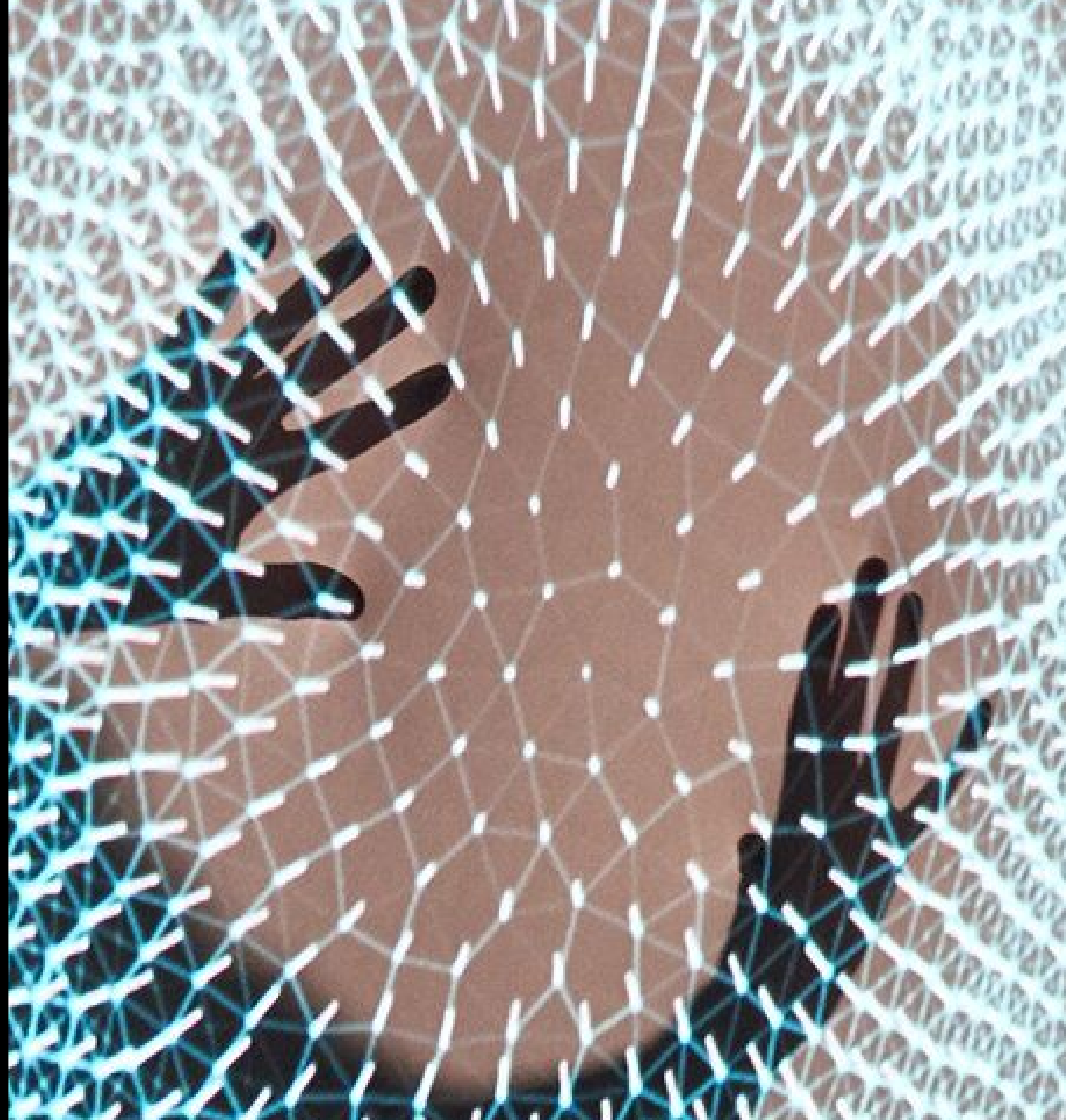
# AAASeed

## 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 On/Off

- **F2**  
→ Toggle **GaBuZoMeu UI**

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

- **F3**  
→ Toggle **render process** On/Off  
**BU\_Alive** visualize this 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  
unused for now (don't worry we working very very hard on it)

ZO

Module Editable Unit : an encapsulated functional Block

MEU

Module Unit : a compact MEU view used to control the execution order

MU

**MOUSE**

Left Button

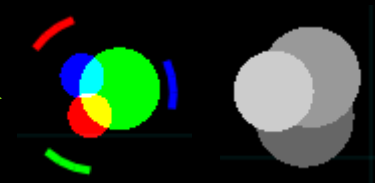
Action

Wheel Button

Scroll

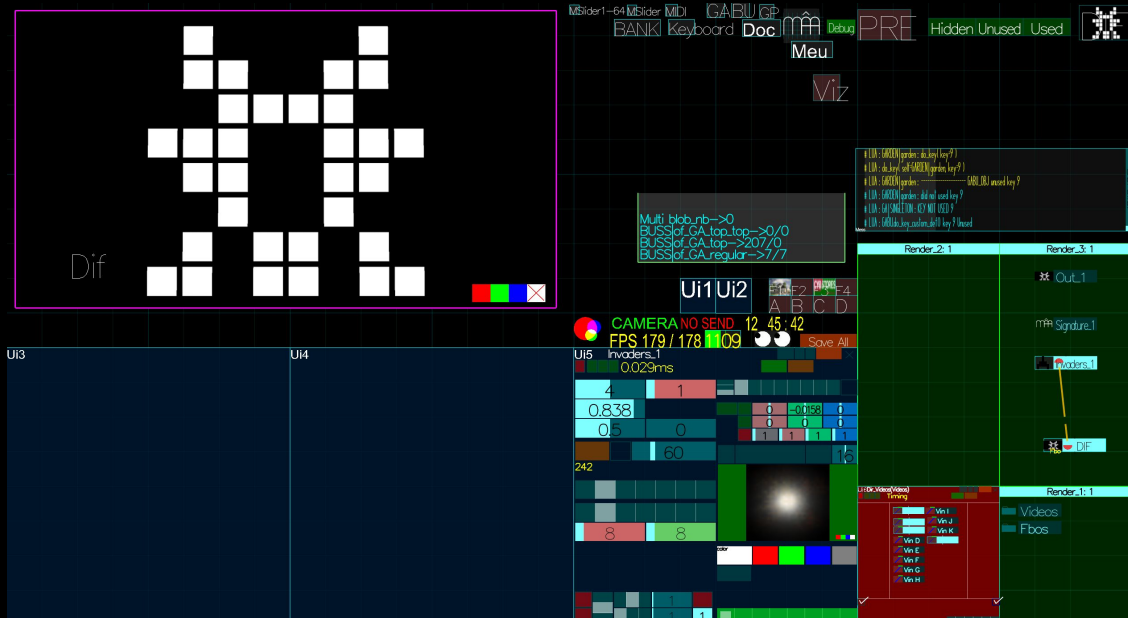
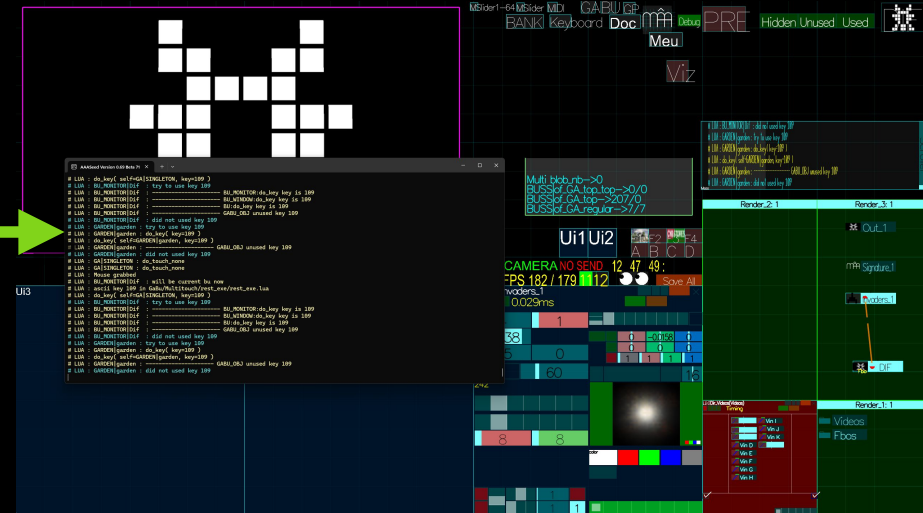
Right Button

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

BUS / BU / BUS / BU...

## Interactive:

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

- To see this

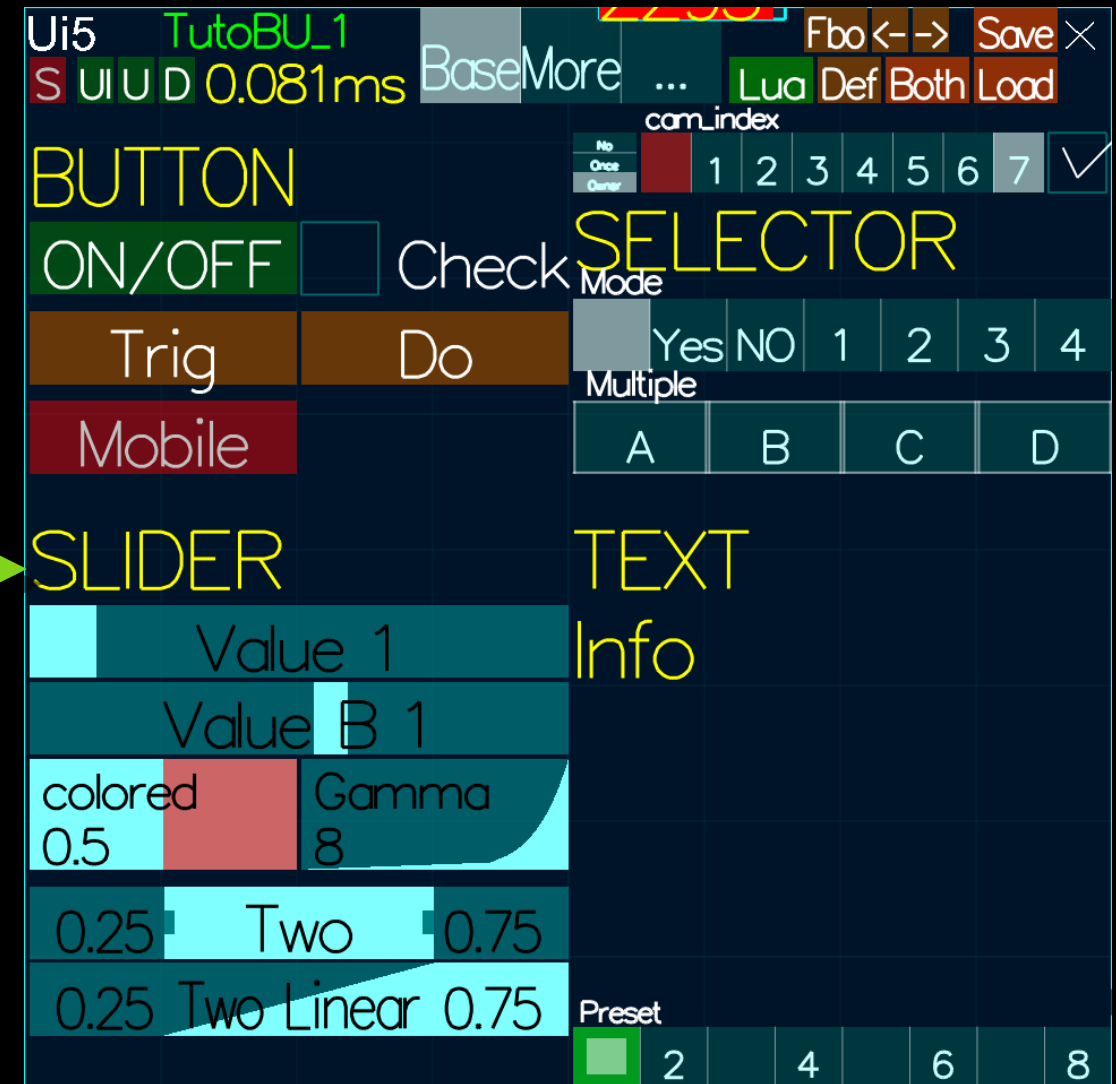
Click on the icon of TutoBU\_1

- Click

Done with **left button**


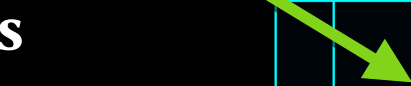
4 types

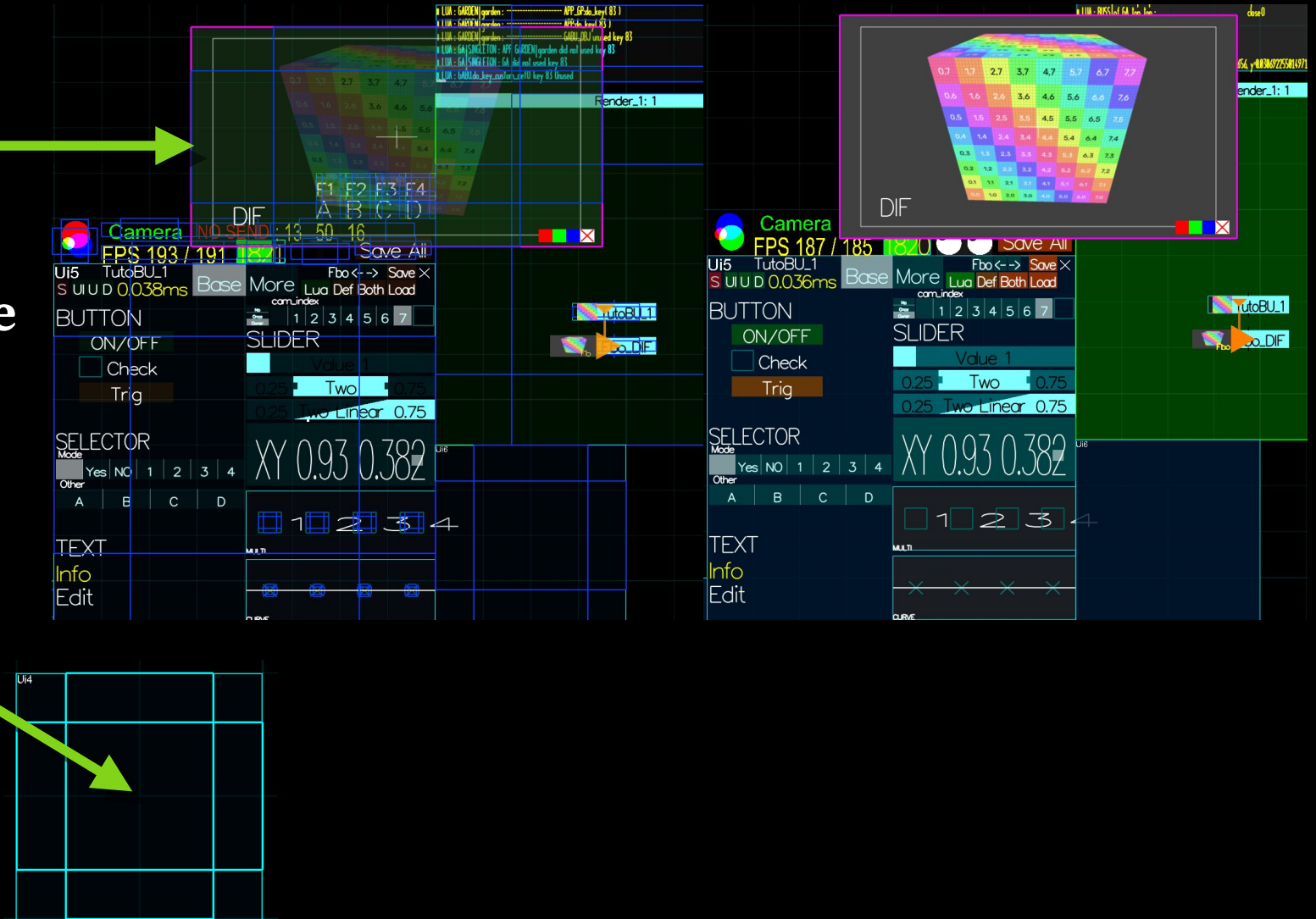
- **Simple**
- **Double**
- **Triple** (rare)
- **Long** (stay fix with button pressed, rare)



# BU Resize Move

- **Hold Alt**
  - **Transparent** 
  - **Blue Flashing Frame** 

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



# BU StarMenu

- **Example:**

Click on TutoBU\_1 slider  
keeping button pushed



- **StarMenu**

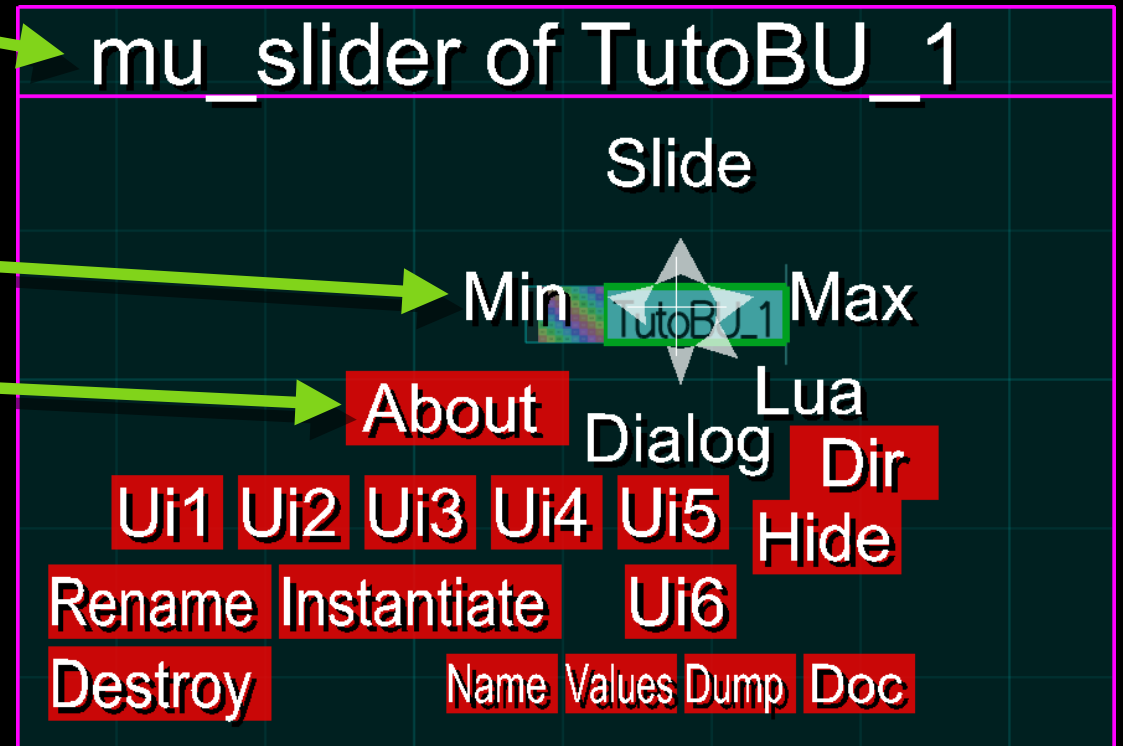
Name of its BU at the top

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**

All BU have a StarMenu, but not all BU display their StarMenu on Click.

In this case, click, and with the mouse button still pressed, go quickly out of the BU rectangle: StarMenu Appears.

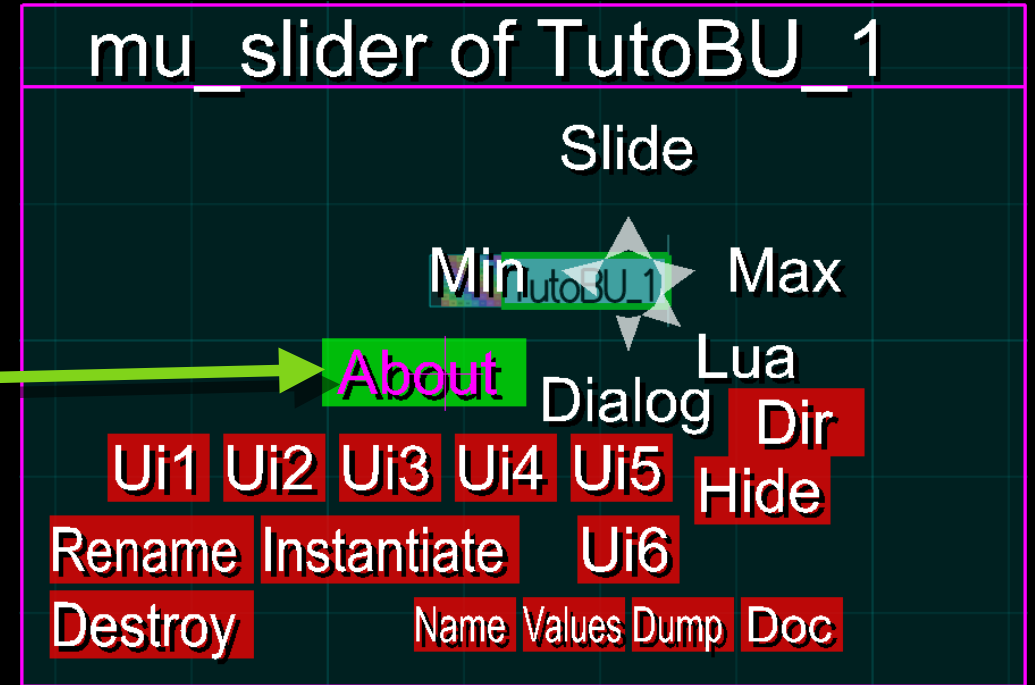


# Dialog

- Example:

**Click on TutoBU\_1 slider**

use **About** button



- **Dialog**

Flashing Red for attention

**Move / Resize using Alt as Always**

**Click to dismiss**

Some ask for confirmation



# Current BU

- Clicked BU

becomes **current**

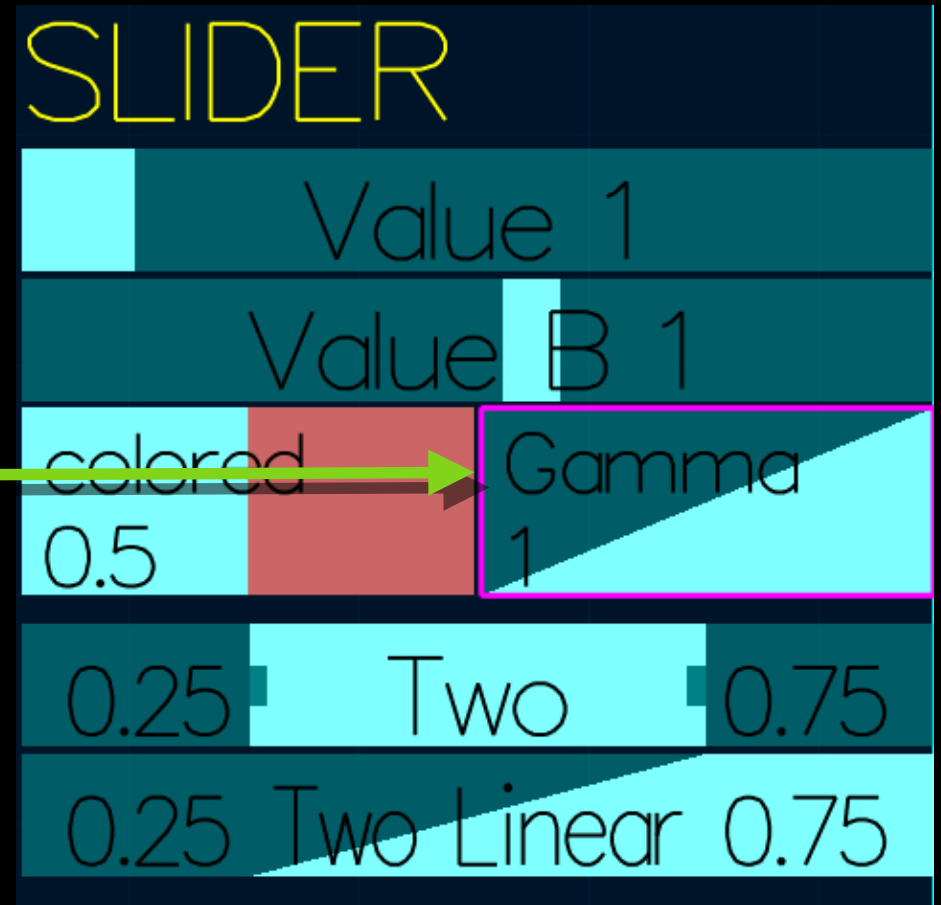
**Violet frame**

**Receives keyboard input**

- **Arrows key to navigate**

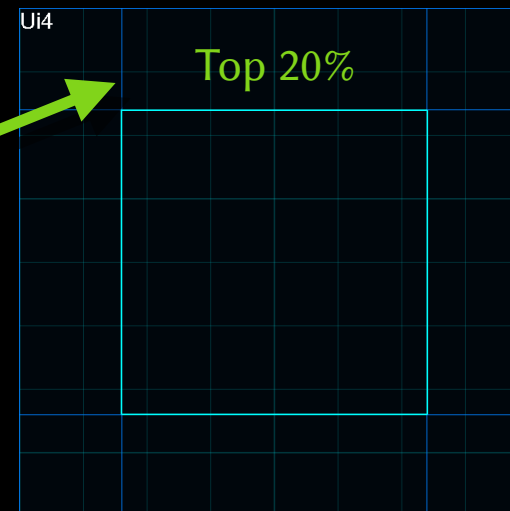
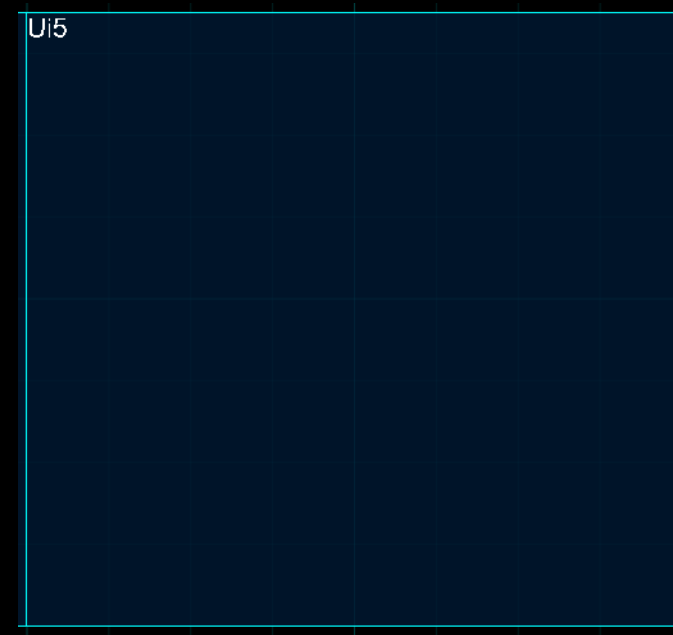
perhaps some bugs left

we need feedback (2025 August)



# BU Resize more

- **Alt click move resize**
  - **Alt Arrow**  
→ **Move**
  - **Alt + - \* /**  
→ **Size**
- **Some BU have 3 states (like a window)**
  - **Normal**
  - **Full** (Big like the screen)
  - **Mini** (Minimized like an Icon)
- **Shortcuts**
  - **Double click, Space**  
→ **Toggle Normal ↔ Full**
  - **Double-click in top 20%**  
→ **Mini**
  - **Ctrl Space**  
→ **Toggle Normal ↔ Mini**



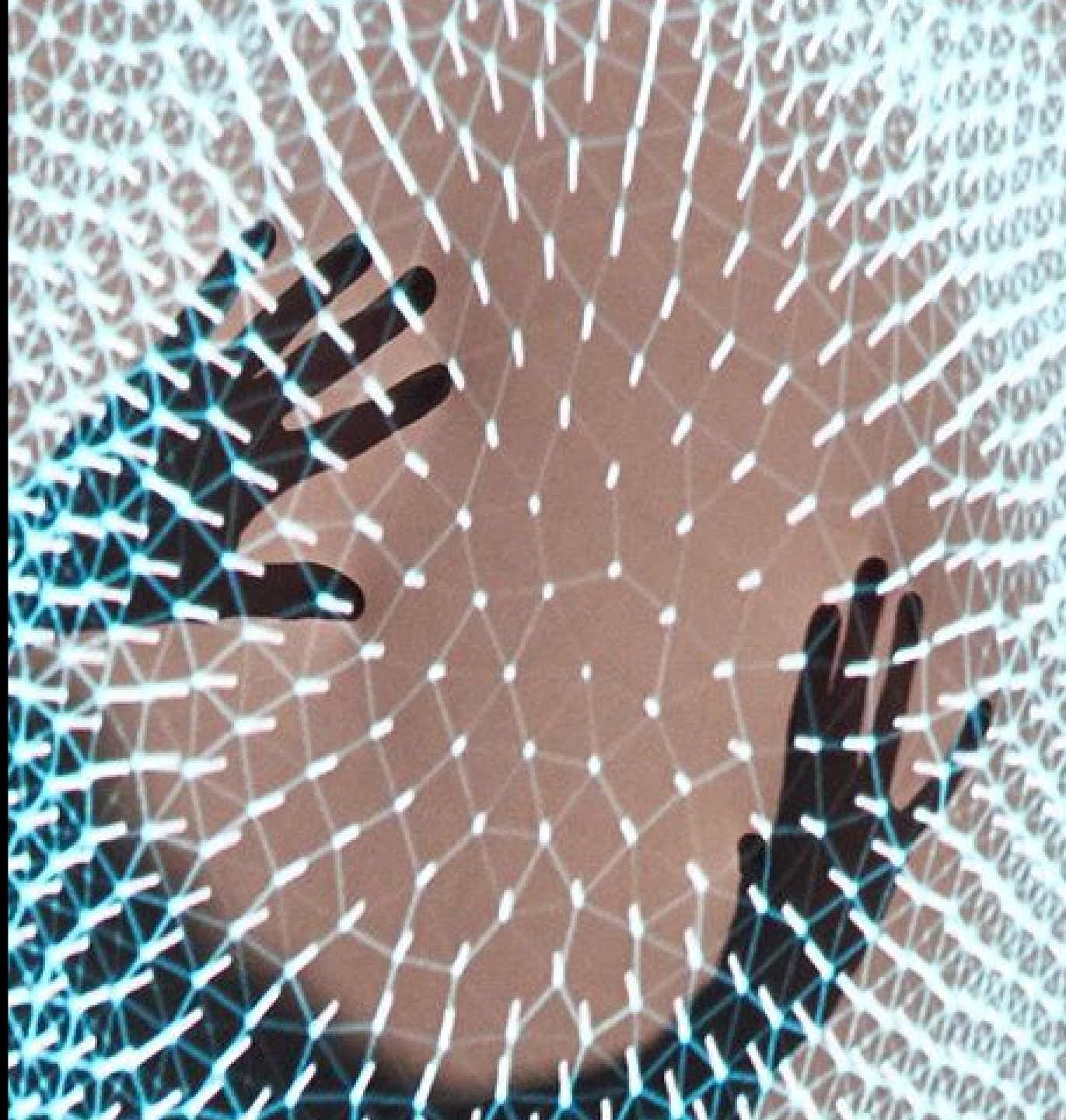
# AAASeed

## An introduction

### Part 5:

# BU\_MONITOR

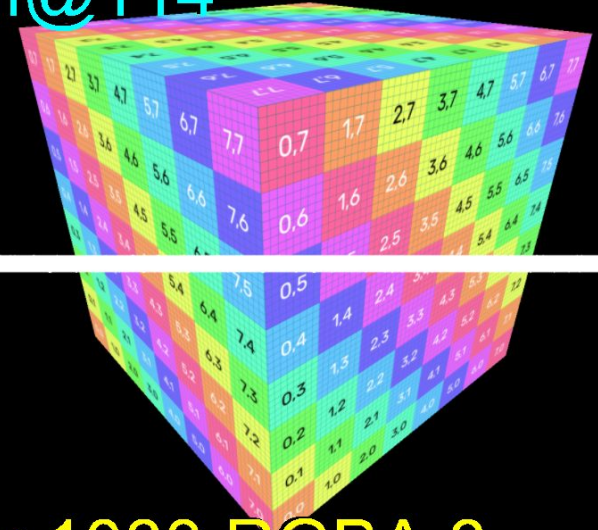
- Introduction
- Channels
- Alpha
- Click Zones
- StarMenu



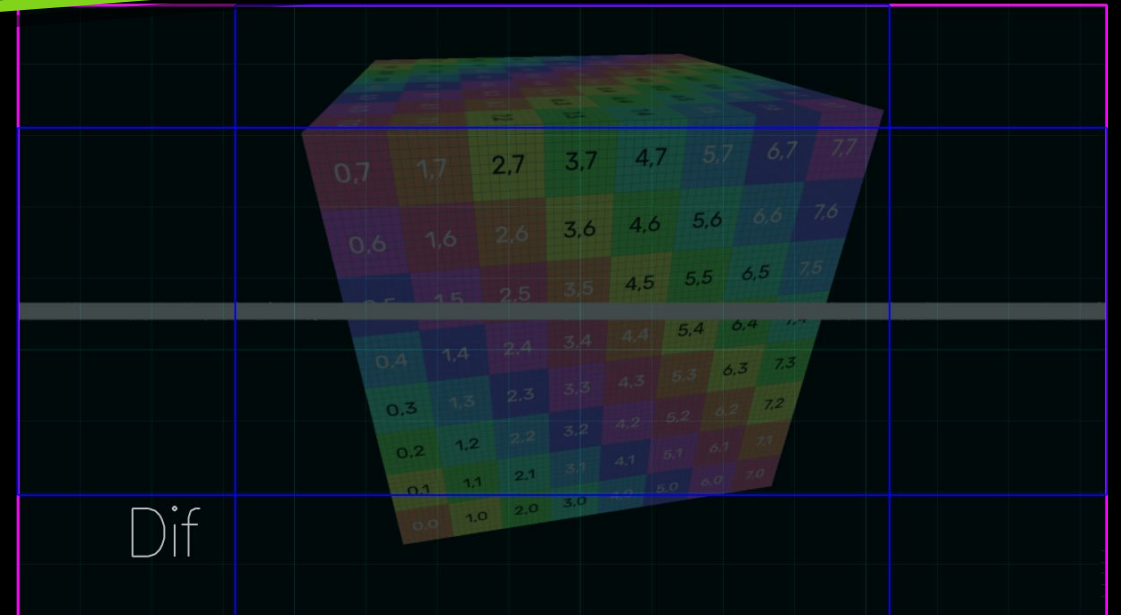
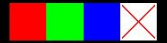
# BU\_MONITOR

- Display Image
- When clicked
  - \_ Display information on image
    - cyan text at top left
      - texture name / location
    - yellow text at bottom left
      - texture format
    - RGBA at bottom right
    - channels used 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...

Dif : Dif@114



1920 x 1080 RGBA 8



# 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

1920 x 1080 Red 8



1920 x 1080 RG 8



1920 x 1080 RGB 8

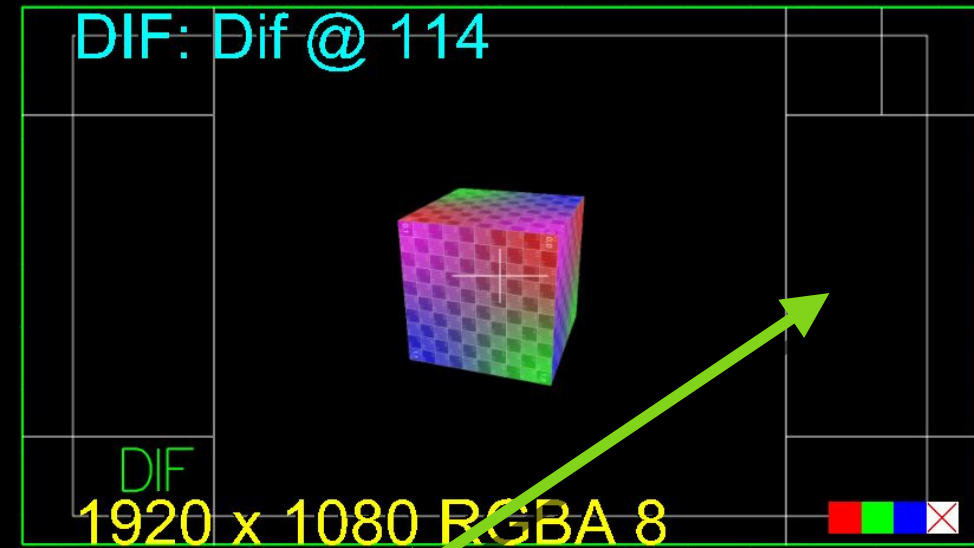
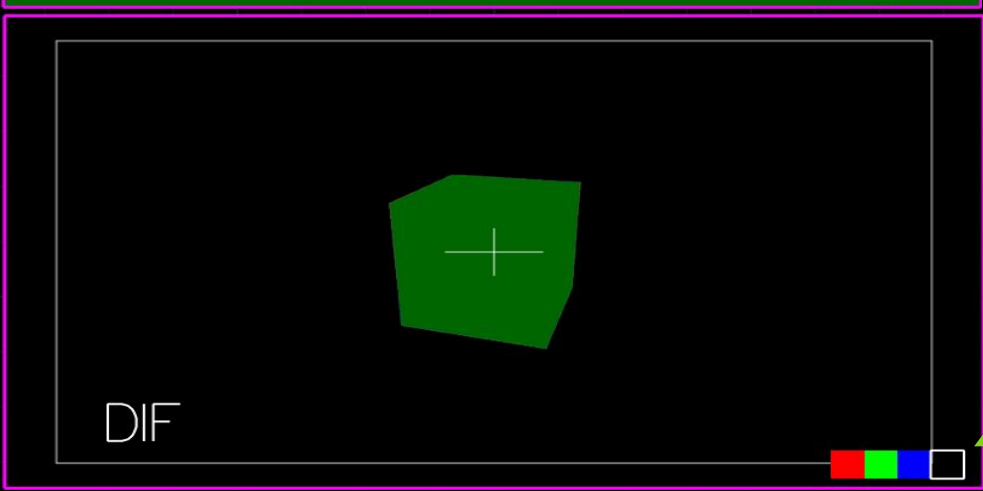
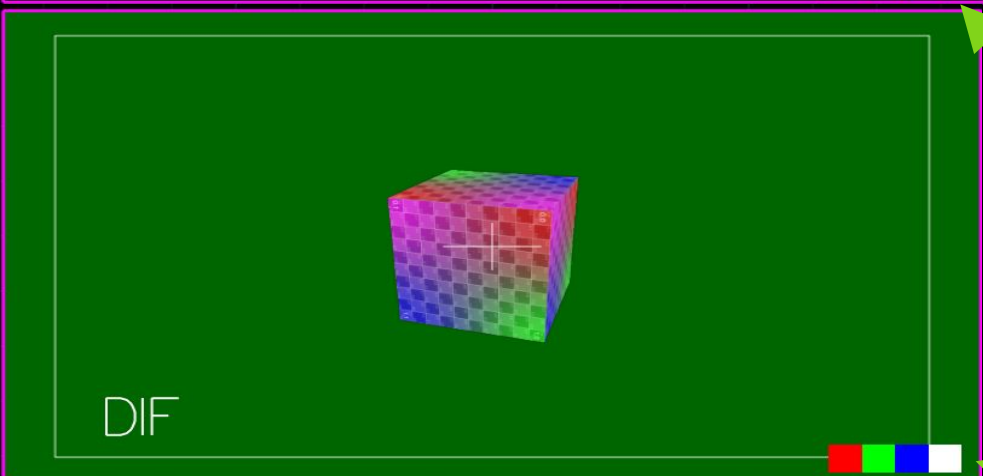
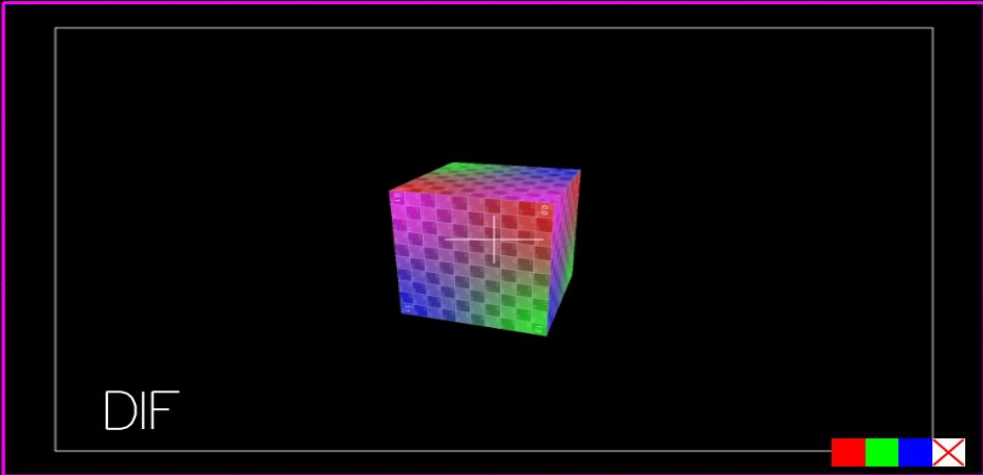


DIF  
1920 x 1080 RGBA 8





# BU\_MONITOR Alpha



- Click in Right Middle or Key + -
  - change how alpha is displayed
  - Ignored
  - Used
  - Used inverted
  - only for RGBA (Only mode with Alpha)

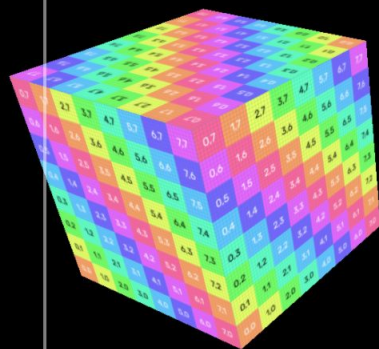
# BU\_MONITOR Click Zones

When the click is released the command is selected and executed

Flip  
Texture  
Bank/Bind  
information

DIF:Dif@114

Flip  
Texture  
informations



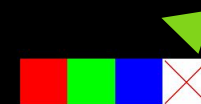
Flip  
Texture  
information:  
resolution and format

DIF  
1920 x 1080 RGBA 8

Flip  
Window  
More

Flip  
Window  
Draw Top

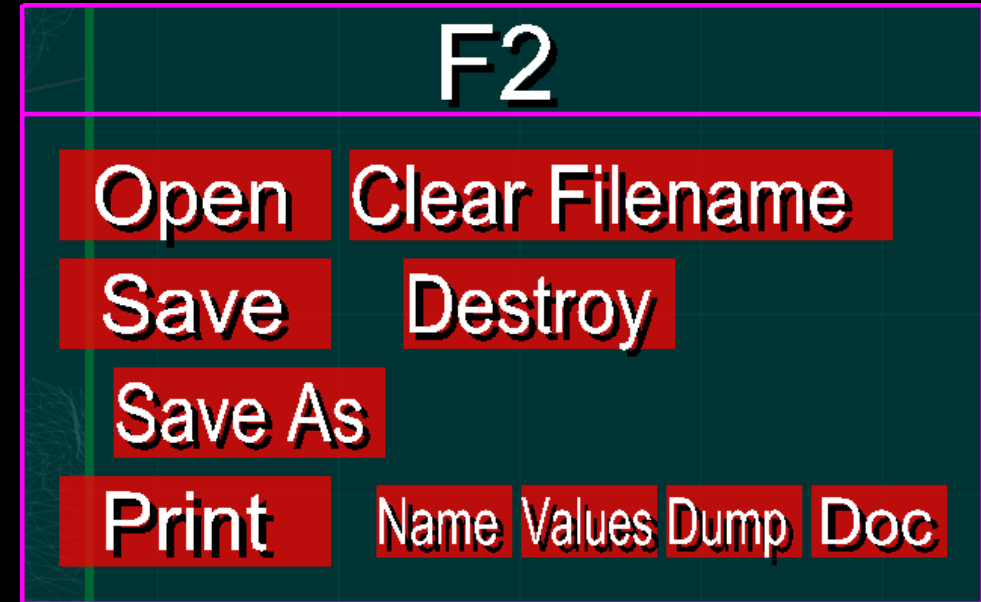
Flip  
Alpha Mode  
also Key + -



Flip  
channel  
display

# BU\_MONITOR StarMenu

- Use **Quick drag-out** like any BU where it doesn't appear directly
- **Open**
  - Load image(s) via a file dialog
    - at the current bind
      - later on this (Part 14: Texture)
    - can choose several files
      - loaded in successive binds
  - Set the filename for this bind
- **Save**
  - Save the Image to Media/AAASnapshots
  - How it is saved is 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 is in Flatland Prefs/MASTER/Print
- **Clear Filename**
  - Clear the the filename for this bind
- **Destroy**
  - Destroy the texture for this bind
  - Clear the filename too: no more pixels or filename left





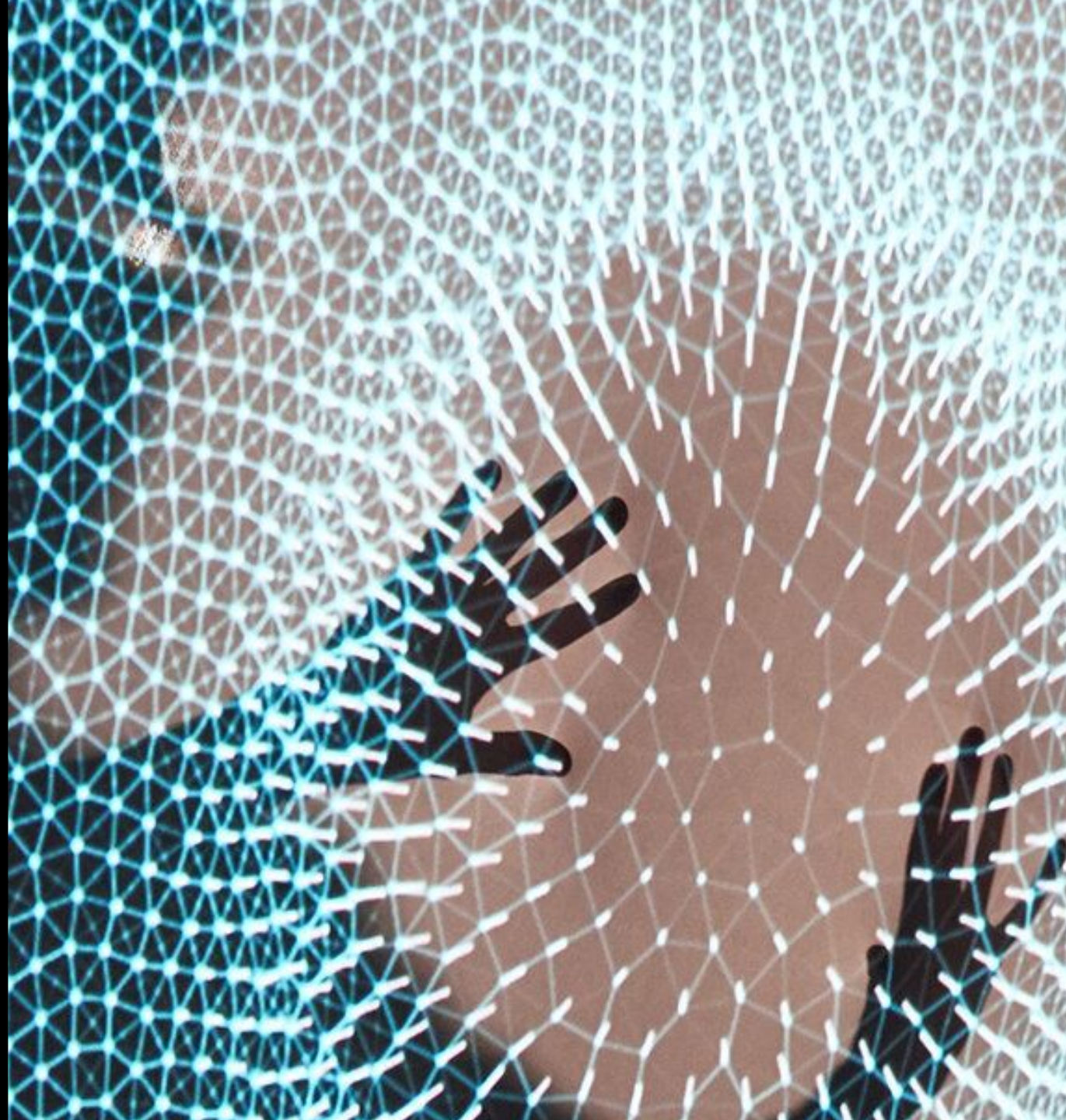
# AAASeed

## An introduction

### Part 6:

## BU is all you need

- Basic BU
- BUTTON
- SLIDER
- SLIDER\_TWO
- SELECTOR
- BU\_TEXT

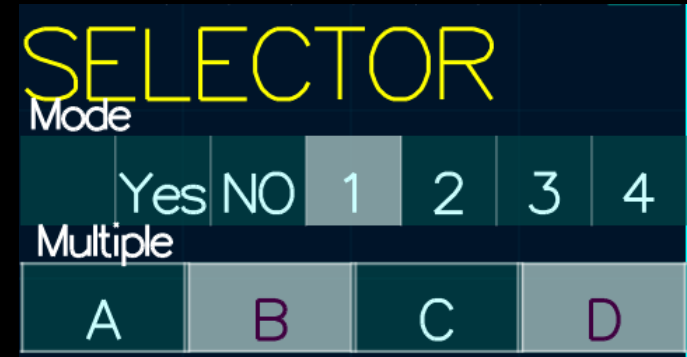
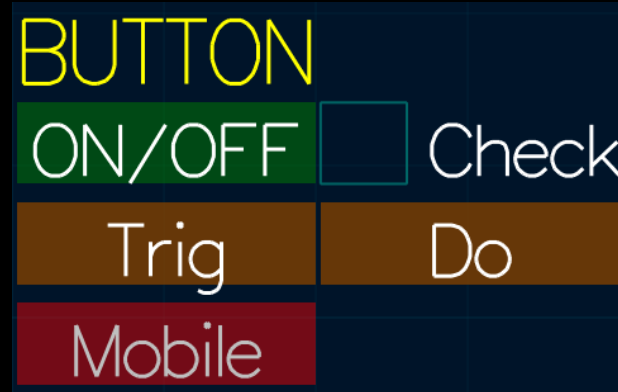


# Basic BU

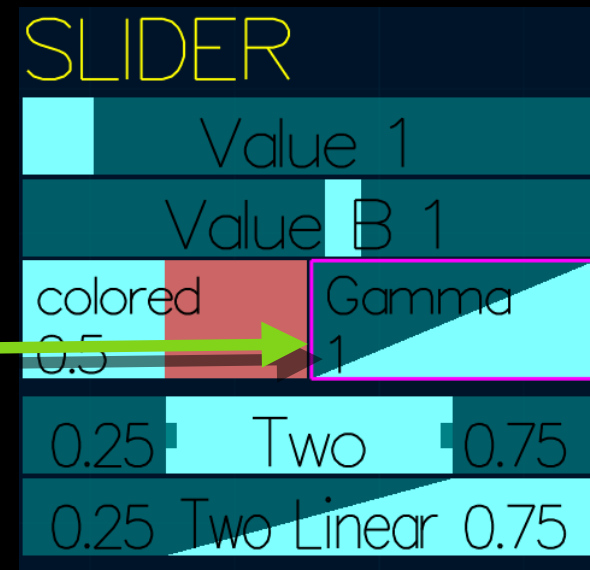
- Click on Train\_BU icon



- BUTTON**
- SLIDER**
- SELECTOR**
- BU\_TEXT**



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

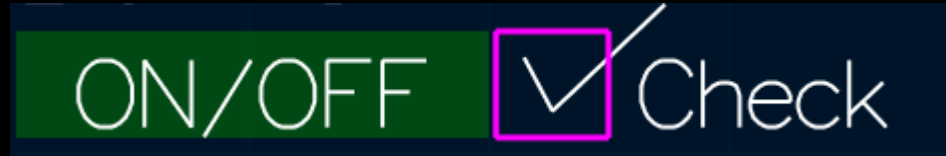


# BUTTON

**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

Try drag the Mobile button



**StarMenu Access**

**Quick drag-out**

**except on mobile BU**





# SLIDER

- Adjust a single value
- Click
  - StarMenu
- Slide

Go up in Slide direction

Go to external circle

External circle → change

Internal circle → no change

Intermediate circle → interpolate

Angle change value

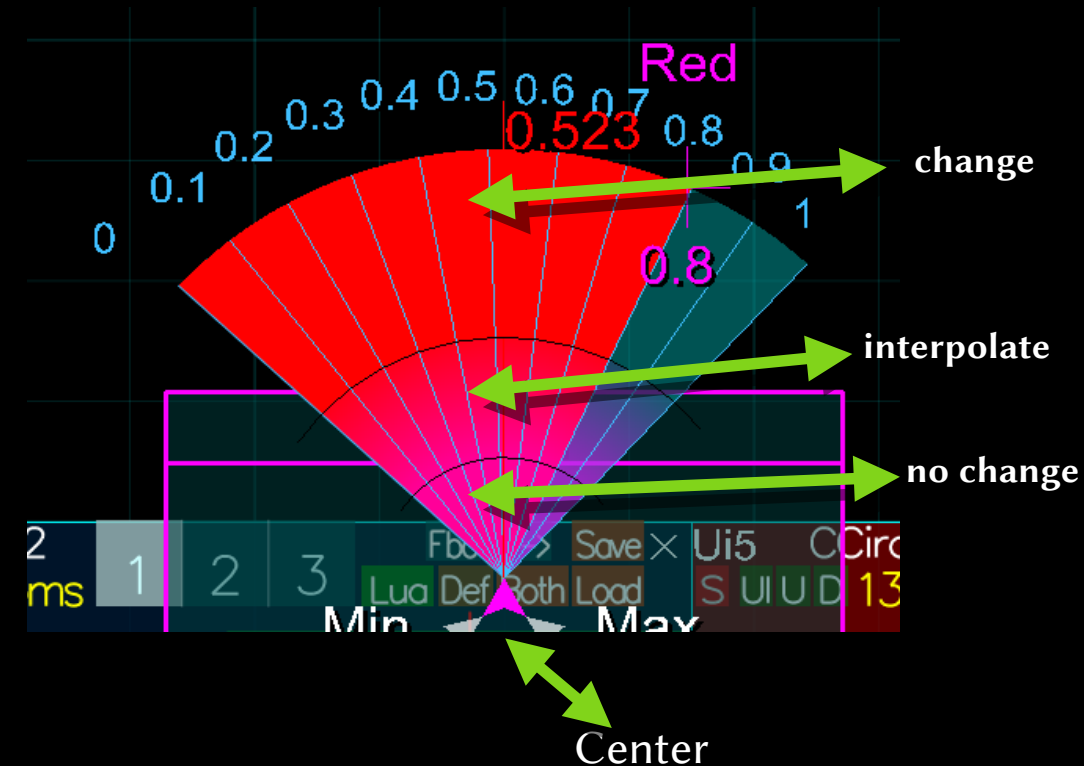
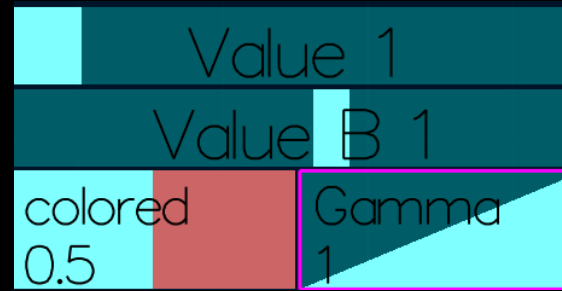
the more away from center the more precise

Go back to internal circle to cancel


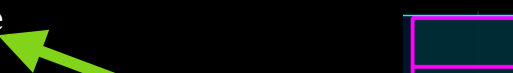
Shift → lock on closest default value

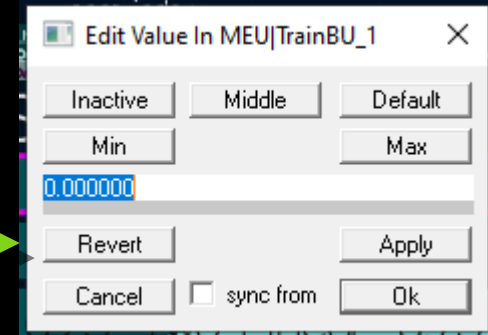
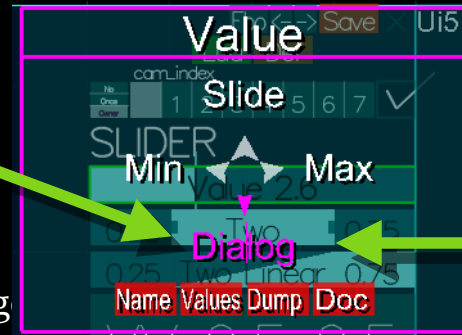
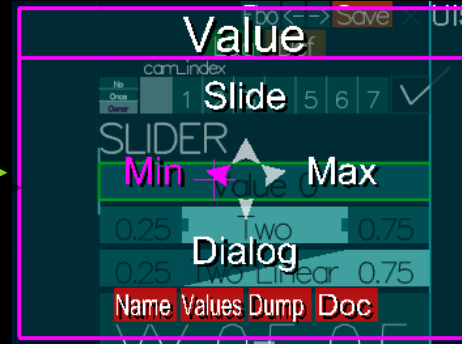
Ctrl → smaller change from start value

Ctrl Alt → even smaller change



# SLIDER

- **Min / Max** 
  - set to Min or Max values defined for each slider
- **Dialog edit**
  - Double Click or Dialog zone 
- **Keys** (act on current BU if no Flatland)
  - + - \* / → **change**
  - **Enter** → **inverse**
  - . → **floor integer** (round to inferior integ
  - **Home** → **Default**
  - **End** → **Inactive**
  - **PageUp** → **Maximum**
  - **PageDown** → **Minimum**
- **slider value** can be
  - clamped to interval [Min,Max]
  - or free from [Min,Max] limitation, in this case Min and Max as just used to display the slider and by the Ui
- **slider value** can be Floating point or integer



# SLIDER\_TWO

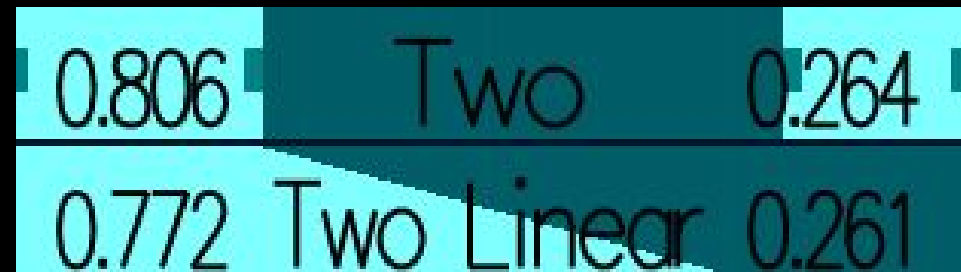
- SLIDER but **2 values**
- Display as Range or Linear
- Value changed depends **where you click**

Closest is selected for interaction

Last interacted value receive keyboard input



- Values can be **Flipped**  
Left value bigger than right  
Displayed differently



# 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

note 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**

- .

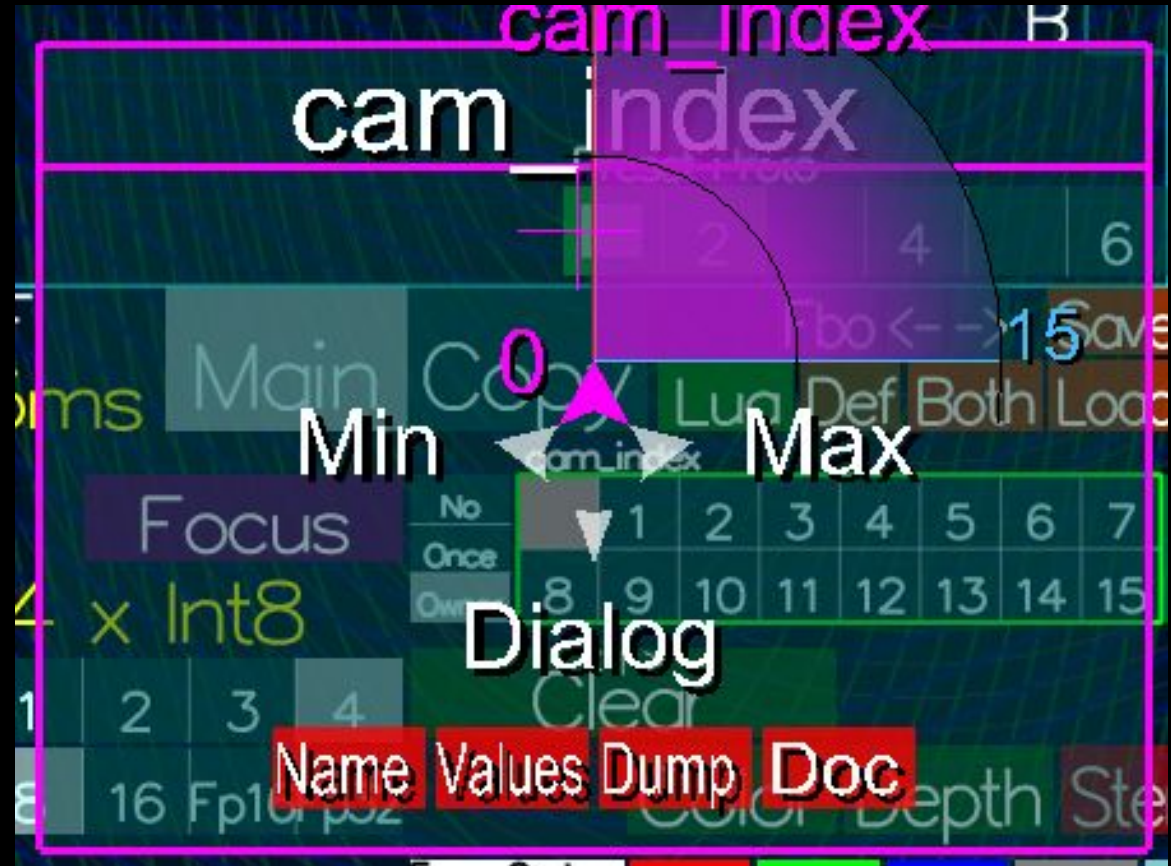
→ **floor integer**

- **Page Down / Up**

→ **Min / Max**

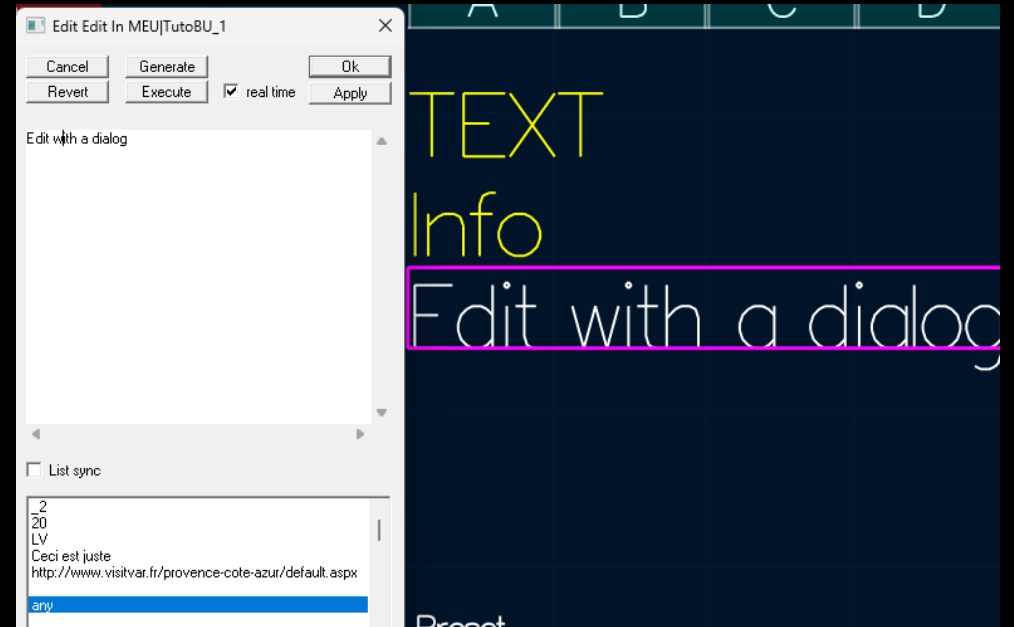
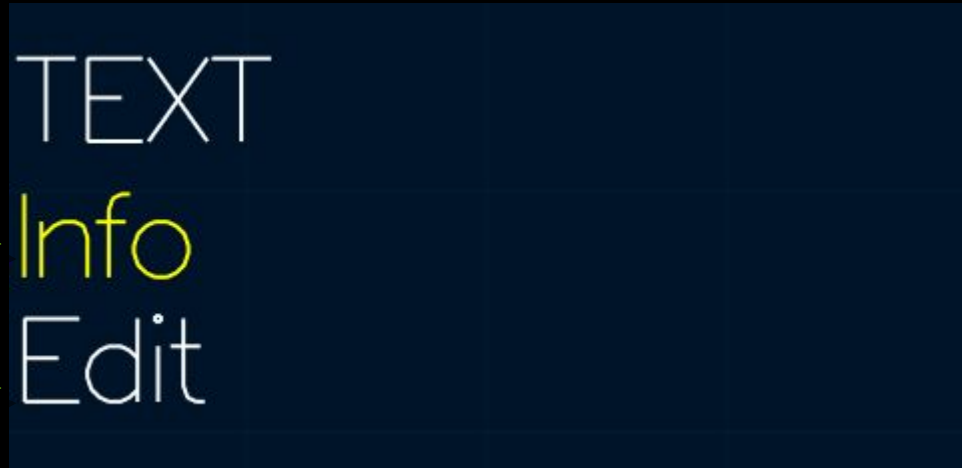
- **Home / End**

→ **Default / Default inactive**



# BU\_TEXT

- **BU\_TEXT info**  
Yellow in general  
display info, not editable
- **BU\_TEXT Editable**  
White in general  
Double Click → Edit
- Access to **StarMenu**  
like **BUTTON** and **SELECTOR**  
Quick drag-out
- will Evolve
- Edit dialog will Evolve too
- Virtual keyboard exists but not functional  
at the moment (2025 August)





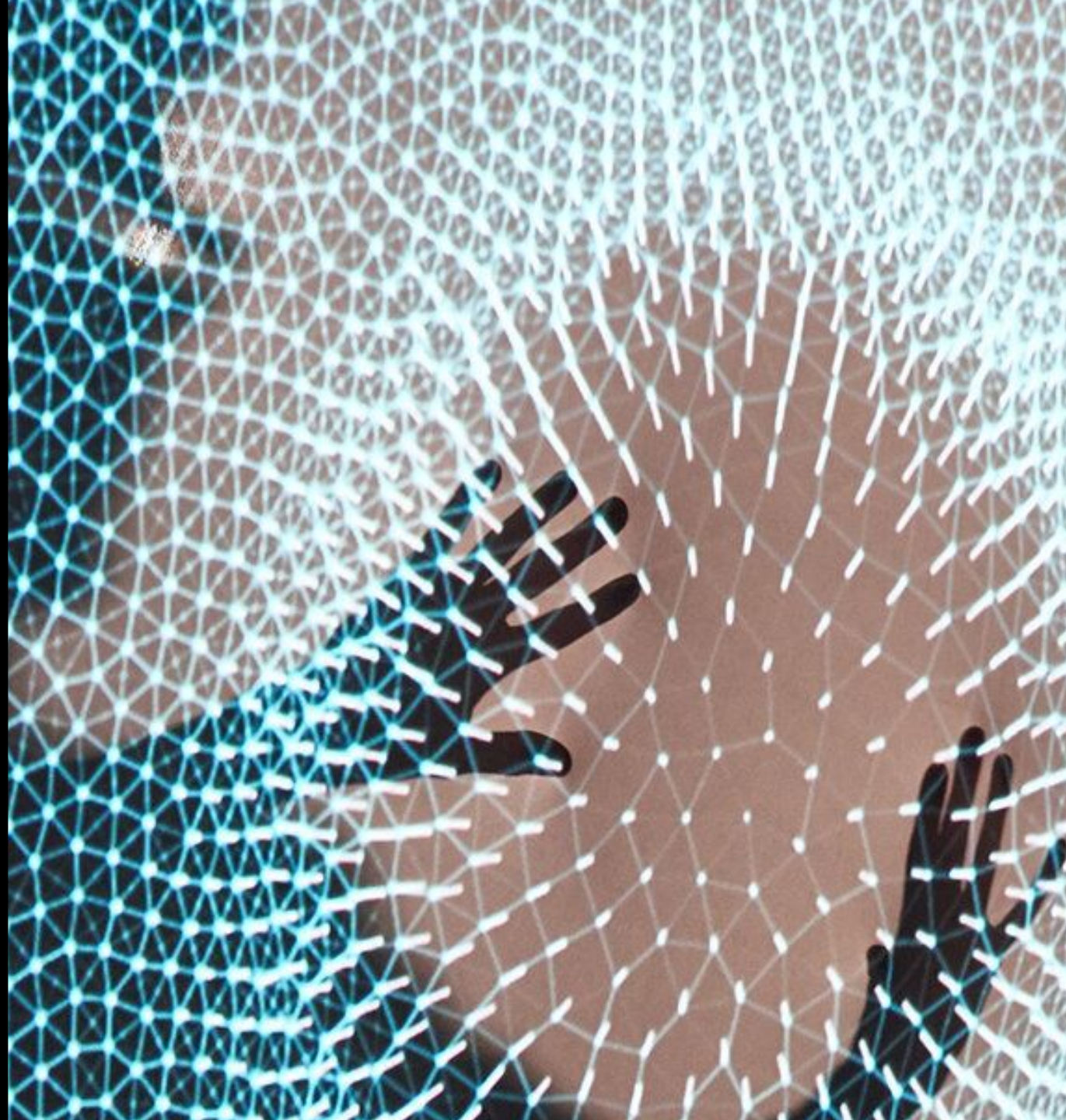
# AAASeed

## 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/Resiz

Alt Arrows → Move

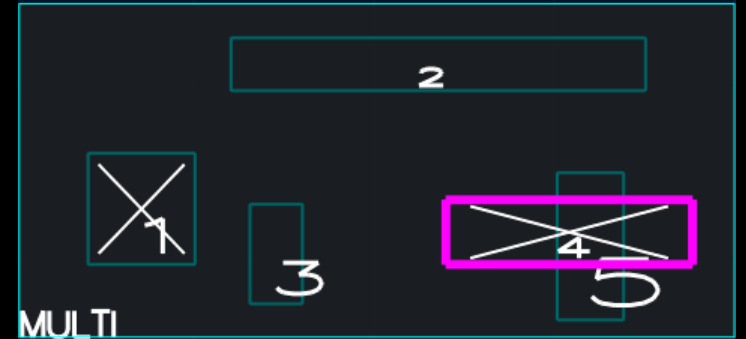
Alt + - \* / → Size

Double Click or Return key

→ toggle On/Off

Key insert → Add point

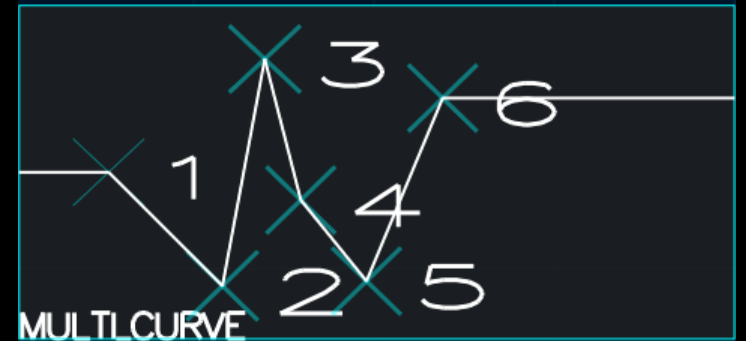
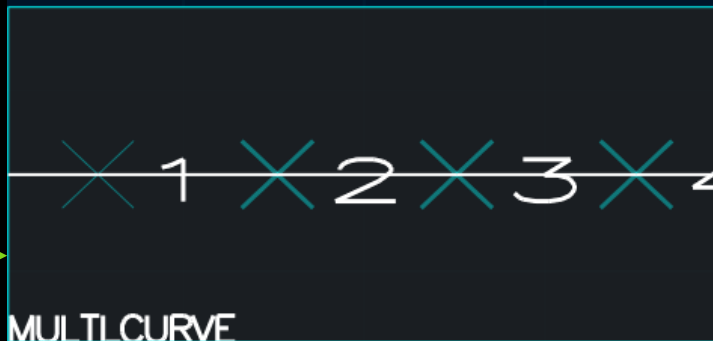
Key delete → Remove point



SLIDER\_MULTI\_CURVE

linear curve

UI identical to SLIDER\_MULTI



# Composite BUs

- **BUTTON** multiple

< | > at the top

Use a hidden **SELECTOR**

**Click left or right** → **change value**

**Same keys as selector**

Option A

Où

Non

- **BUTTON** menu

small rectangle at the top

**Click**

→ open a **SELECTOR** in a **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**

Add





# 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

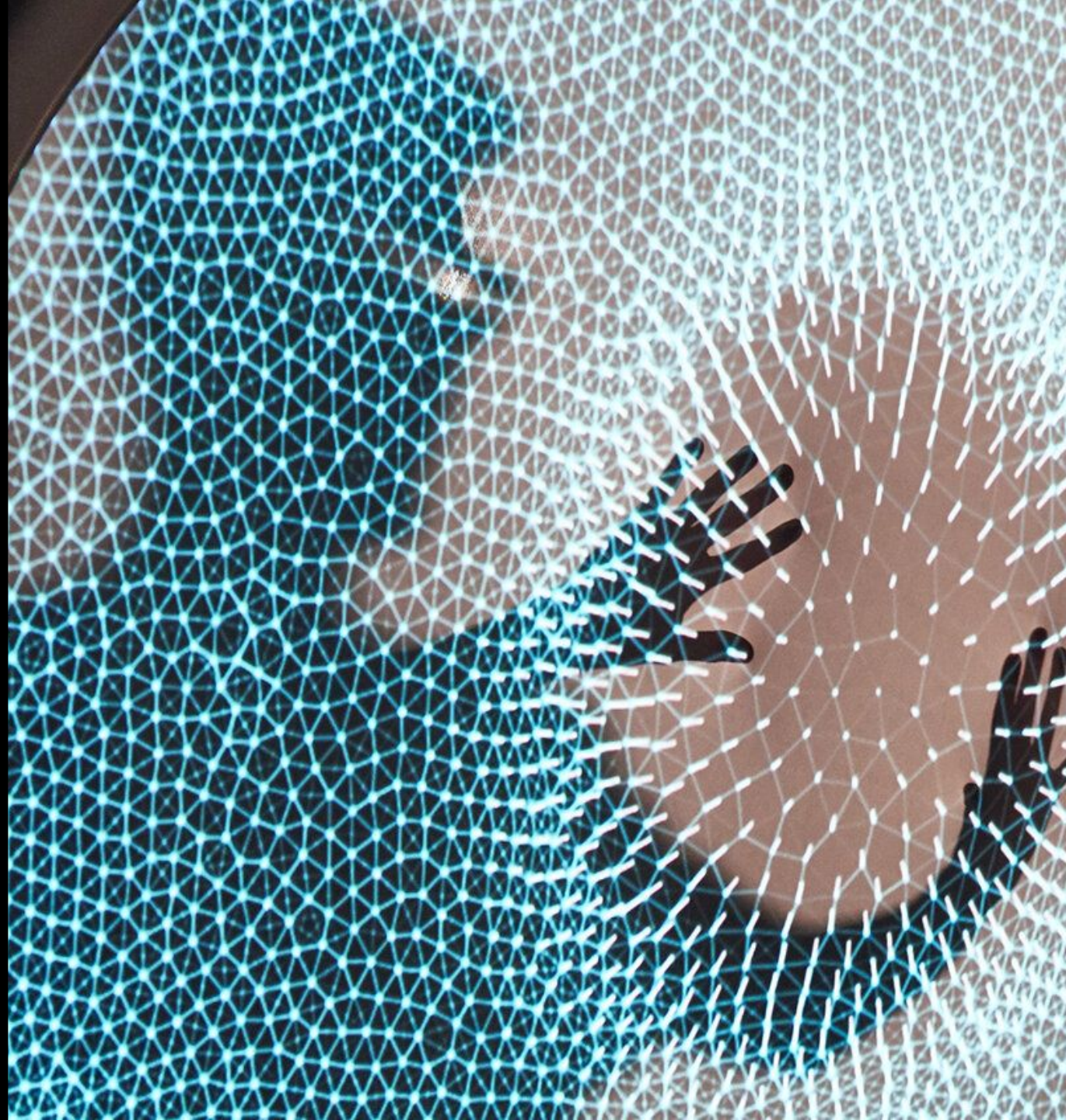
- **Ctrl z / Ctrl y**
- **Deal with**
  - BU values**
  - BU size and position**
- **Unlimited**
- **One more reason not to be afraid**
  - try things then cancel**
- **Shift Paste Special case (2025 August)**
  - treated as a series of individual actions for now**
  - instead of a global change**
-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

# AAASeed

## An introduction


### Part 8: Specialized BU

- BU\_WIZs (Wizard)
- BU\_MESS
- BU\_SHOW
- Window GA
- Window BU
- Window GP





# Some BU\_WIZ (Wizard)

- **BU\_ALIVE** Toggles rendering On/Off (Click or F3)
- **BU\_FPS** Frame Per Second **FPS 153 / 145**  
FPS also in Flatland (top left): so we can see FPS with no UI
- **BU\_TIME** Shows time **23 : 26 58**
- **BU\_EYE** Shows mouse direction 
- **BU\_CAM** Manages camera editing **CAM Locked**
- **BU\_SEND** Controls Ui synchronization between machines  
**NO SEND**
- **BU\_MEM** Shows memory usage in megabytes **1165**  
a problem if it increase continuously
- **BU\_POWER** Displays power and plug status **Power : Plugged 92%**
- **BU\_BLOB** Shows 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
```

Mess

Scroller

Displays system messages and errors

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

```
# LUA : { dialog_id2, whatchange_value, value2, b_return1 }
# LUA : aa.dialog_hook "change_value" for BUTTON[what with do_dialog_hook
# LUA : { self=BUTTON[what, whatchange_value, value2, dialog_table=table: 0x47c3d0b }
# LUA : dialog_id2, whatchange_value, value2, b_return1 }
# LUA : aa.dialog_hook "change_value" for BUTTON[what with do_dialog_hook
# LUA : { self=BUTTON[what, whatchange_value, value2, dialog_table=table: 0x47c3d0b }
# LUA : { dialog_id2, whatchange_value, value2, b_return1 }
# LUA : aa.dialog_hook "change_value" for BUTTON[what with do_dialog_hook
# LUA : { self=BUTTON[what, whatchange_value, value2, dialog_table=table: 0x47c3d0b }
# LUA : { dialog_id2, whatchange_value, value2, b_return1 }
# LUA : aa.dialog_hook "change_value" for BUTTON[what with do_dialog_hook
# LUA : { self=BUTTON[what, whatchange_value, value2, dialog_table=table: 0x47c3d0b }
# LUA : Mouse grabbed
# LUA : SELECTOR[menu choose : 1 2
# LUA : BUTTON[what : 1 2
# LUA : Mouse grabbed
# LUA : SELECTOR[menu choose : 2 3
# LUA : do_click_upc self=BUTTON[what, x=1.028830114215, y=0.7911271761698, uiftable: 0x4ab32e0, name_sel=slide )
# LUA : do_self_command self=BUTTON[what, uiftable: 0x4ab32e0
# LUA : print fn mess : GABU_OBJ.do_self_command_with_super
# LUA : do_self_command self=BUTTON[what, b_used=false, uiftable: 0x4ab32e0, class=GUI )
# LUA : BUTTON[what : BU did not recognized command "slide" try now with BU
# LUA : do_self_command self=BUTTON[what, uiftable: 0x4ab32e0, command=slide )
# LUA : print fn mess : GABU_OBJ.do_self_command_with_super
# LUA : do_self_command self=BUTTON[what, b_used=false, uiftable: 0x4ab32e0, class=BU )
# LUA : BUTTON[what : BU did not recognized command "slide" try now with GABU_OBJ.NIL
# LUA : BUTTON[what : GABU_OBJ did not recognized action "slide"
# LUA : BUTTON[what : 2 3
# Sound beep ----- AAASeed has been running for 0 h 13 min 28 seconds
```

# BU\_SHOW (Debug Tool)

```
Multi blob_nb->0  
BUSSof_GA_top_top->0/0  
BUSSof_GA_top->220/0  
BUSSof_GA_regular->7/7
```

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

In Lua, to show name→value line use

```
aaa.show( value, name )
```

```
self:show( value, name ) for objects
```

# Window GA first look

- Global Action 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** by default
  - **Finger**  
What we use in the tutorial videos to **display the mouse position**
  - **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

☒ UIF
 ☐ On Shift

☐ Help show
 ☐ Grid Top

☒ 8 Size 12
 ☐ Menu Time Alive 4
 ☒ BU\_SHOW Time 5

☒ Finger
 ☐ number
 ☒ as lines

☒ Far draw Less
 ☒ Only Mouse

☐ Spy
 ☐ LuaJit
 ☐ LuaJit\_dump
 ☐ Visualization

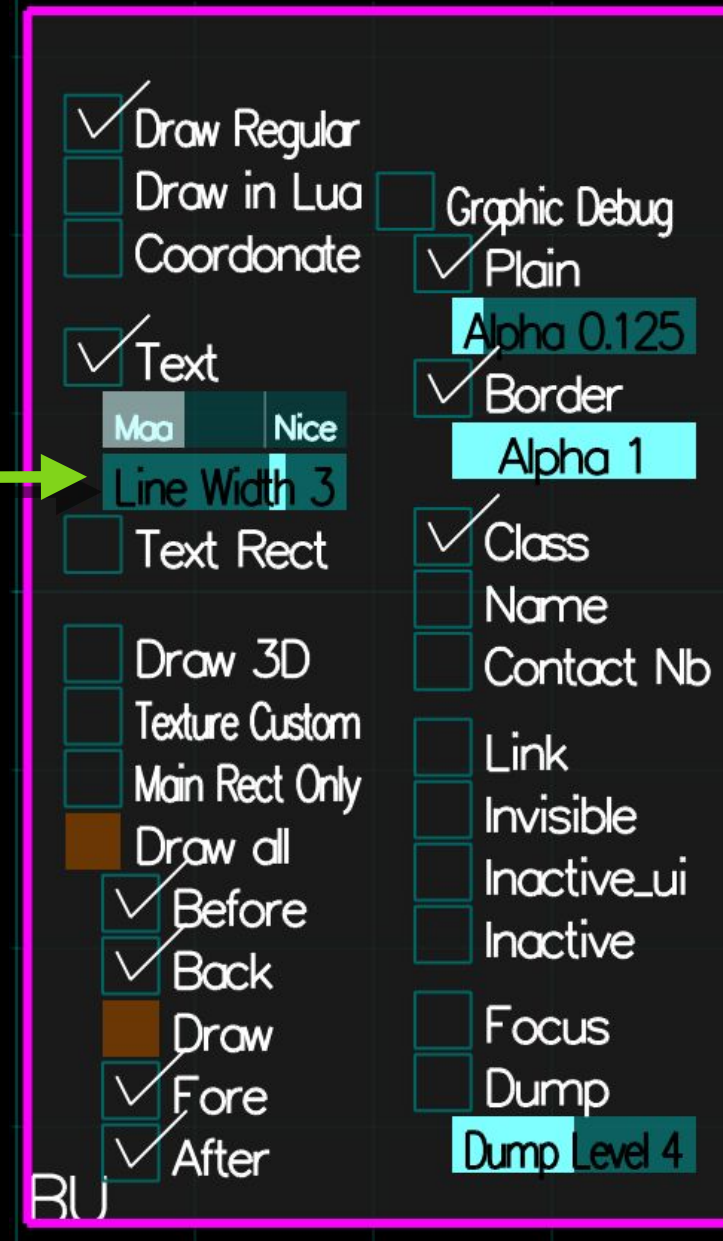
☐ Offset X 0
 ☒ Offset Y 0

Verbose

MEU FILE			
UIF			
GA			
BUSS			
BUS			
BU			
BUI			
BUTTON			
UNDO,REDO			
BLOBS			
BLOB			
EVENT			
VIDEO			
IMGS			
APP			
MUS			
MU			
MEU			
MEDIA			

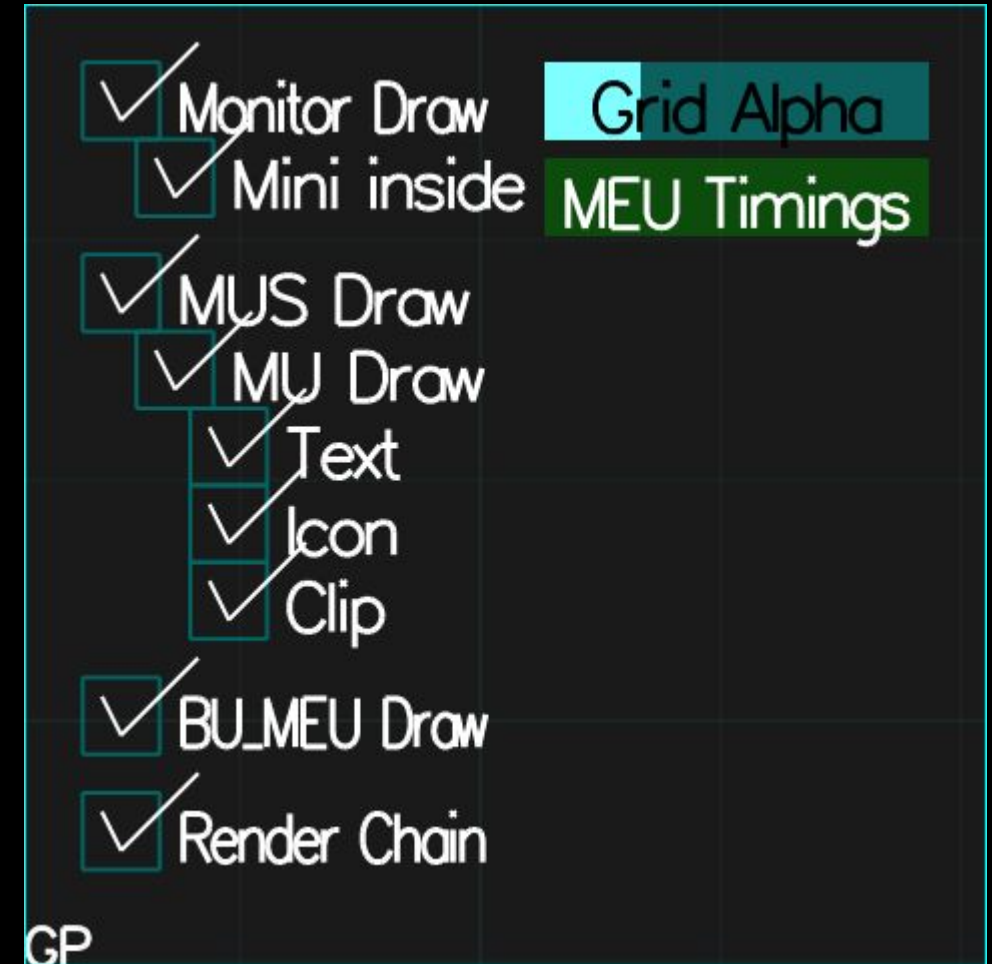
# Window BU

- **Box User 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 Garden Party
  - first time the MEU/MU interface was used, it was during a french company Garden Party (Utram).
- 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





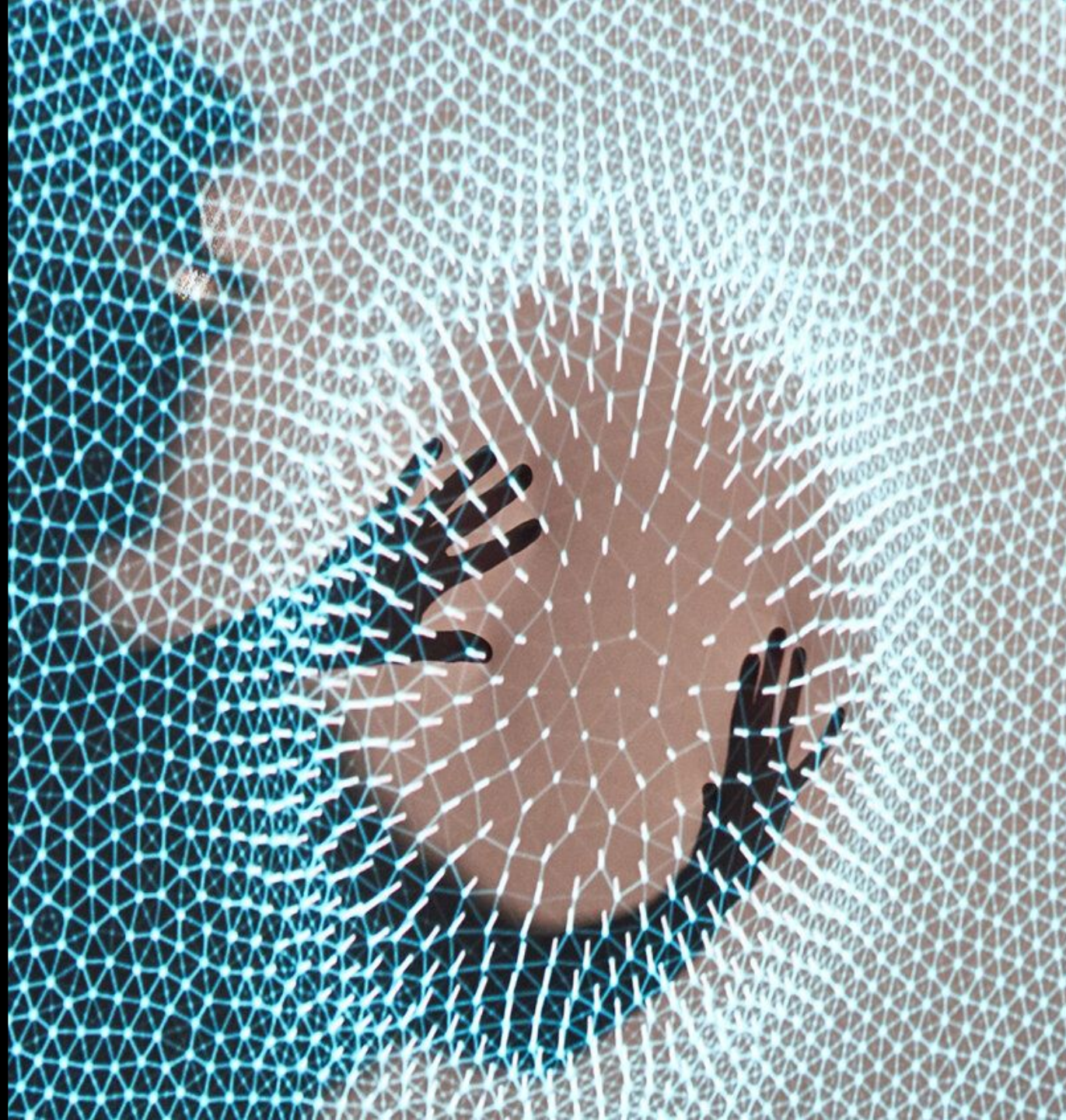
# AAASeed

## An introduction

### Part 9:

### MEU

- MEU Module Editable Unit
- MU Module Unit
- MEU and Uix
- More on MEU
- MEU Bar
- MEU Preset
- MEU StarMenu





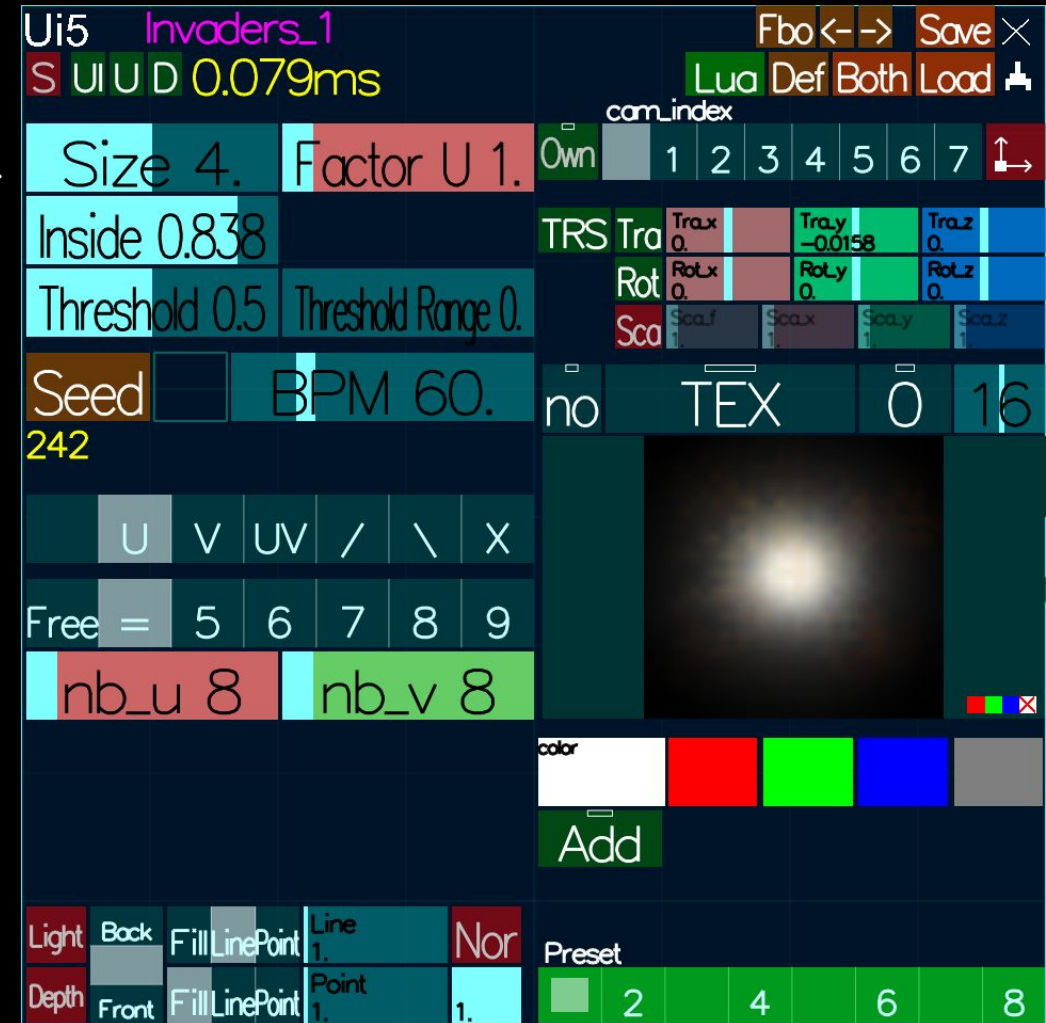
# MEU Module Editable Unit

## MU Module Unit

- The MEU is the core element 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
  - open MEU interface in **Ui5 / Ui6**
    - Ui5** is the default
    - Ui6** is the default for MEU DIRs
- **Link** shows corresponding MEU / MU
- **MU StarMenu** to choose Uix
- **MEU StarMenu**
  - click on background 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 d other tasks:
  - receives/send data**
  - analyses image**
  - control a device** (e.g. plotter, projector, Dmx, Arduino...)
  - ...
- **2 main methods** (functions) 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 parameters 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 of type TutoBU instance 1**
- **S UI UD** **buttons**  
**Switches to activate/deactivate call to methods**  
Send, update\_UI, Update, and Draw
- **0.032ms** **Execution time** in millisecond
- **BaseMore ...** is a **SELECTOR** named **TAB**  
**switch** between different block of MEU **Ui**
- **X** **Close button detach** the **MEU** from its **Uix**



# MEU Bar



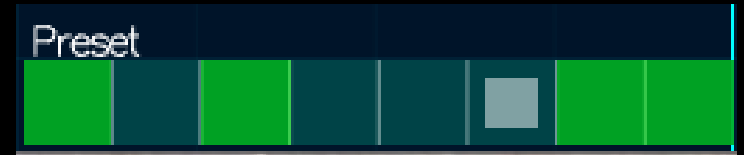
- **Lua** Opens the MEU's lua script in your associated text editor
- **Load** Loads only this meu state
- **Save** Saves only this meu state
- **Def** Trigger a **Definition**, in fact a **redefinition** of the MEU (its Ui in particular)
- **Both** Trigger **Both** a re**Definition** and a **Load**
  
- **Fbo** Open the used **Fbo** (more soon on Fbo)  
in the previous **Uix**
- **<- ->** lets you navigate in the render 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 **BUs**  
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 (2025 August)



Preset							



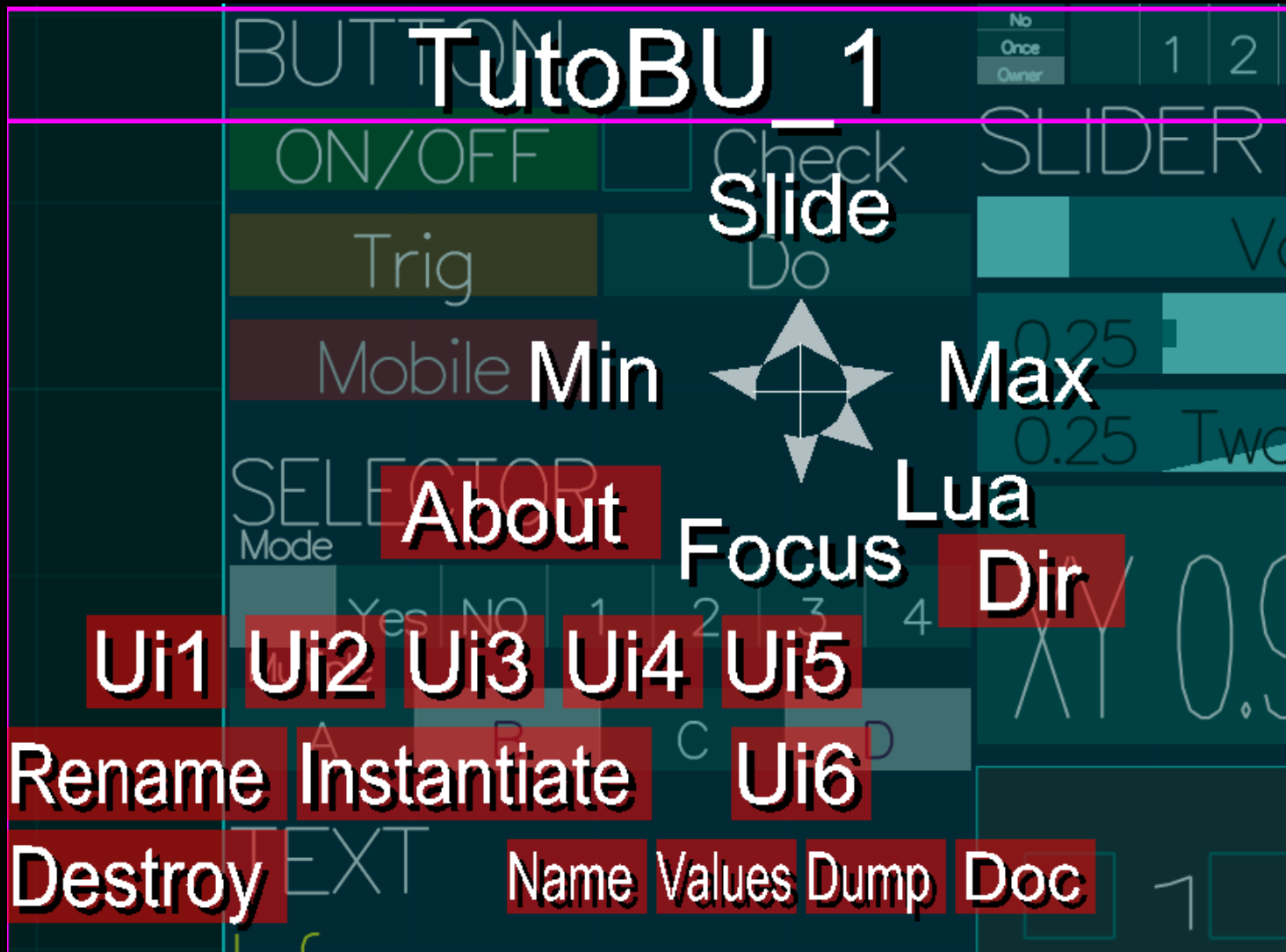
Preset							



Preset							

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 the MEU/MU
    - Type can't be changed



# AAASeed

## An introduction

### Part 10: Render chain

- Render chain
  - **Order**
  - **Alpha / Opacity**
  - **Visualization**
- BU\_RECT
  - **Order**
  - **Value**
  - **StarMenu**
  - **Move**

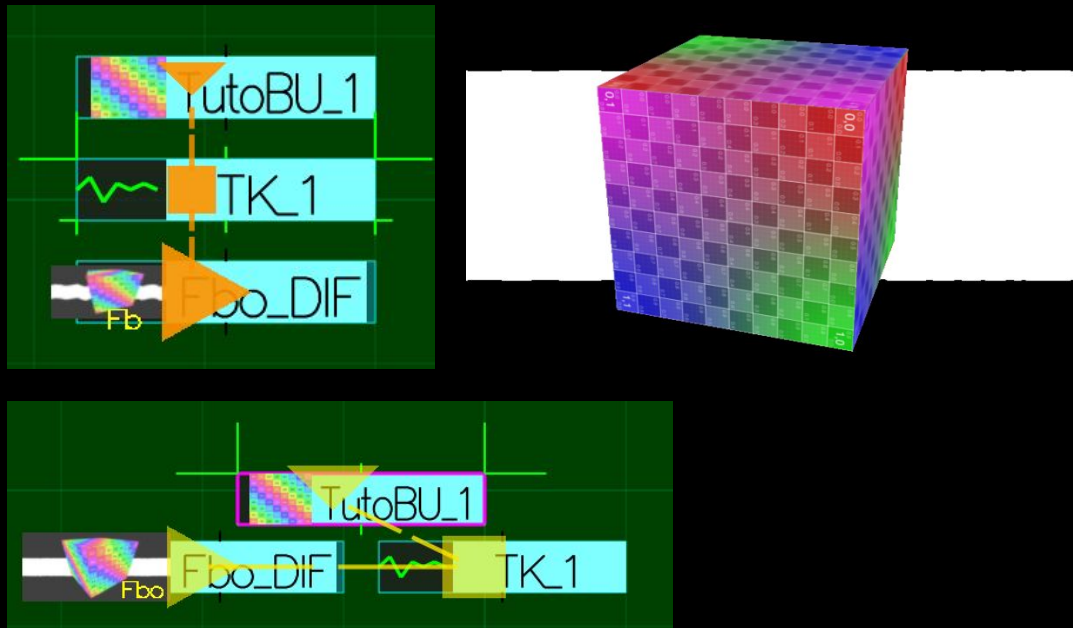




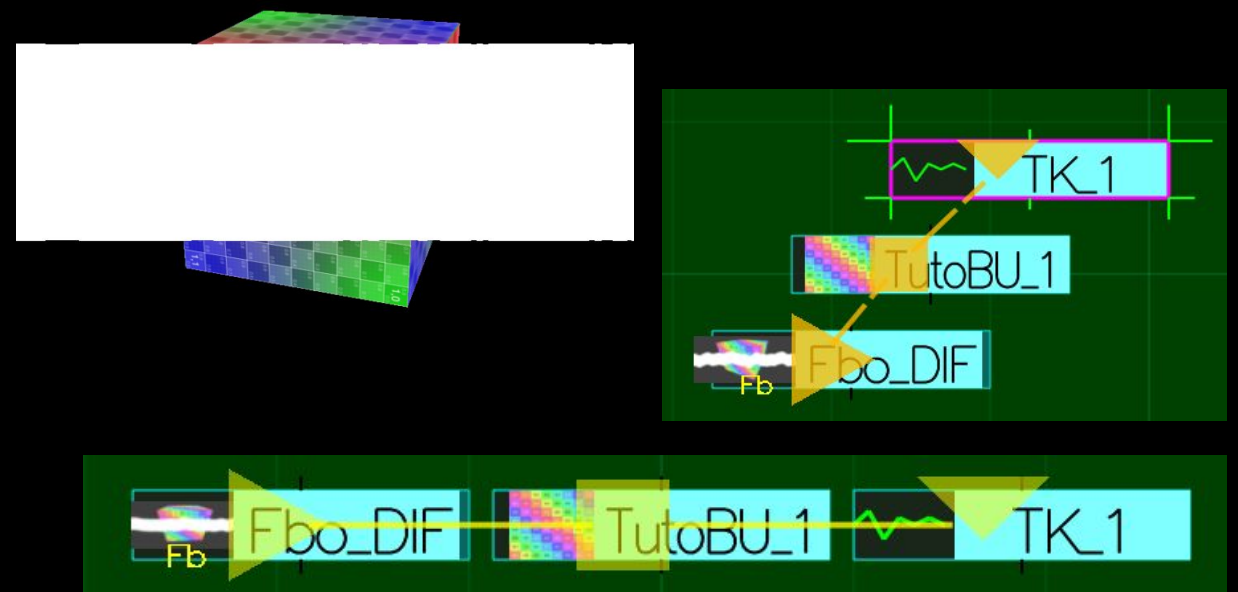
# Render chain: Order

- The **render chain** is the **order** in which all the MEUs in your application are executed.
- The **positions** of the **MUs** control the **execution/rendering order**:  
**bottom to top, then left to right**

TK\_1 rendered first, then TutoBU\_1



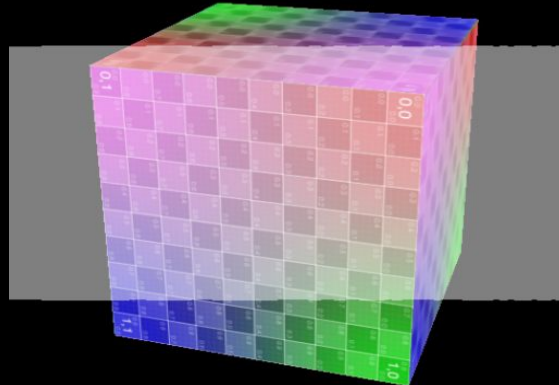
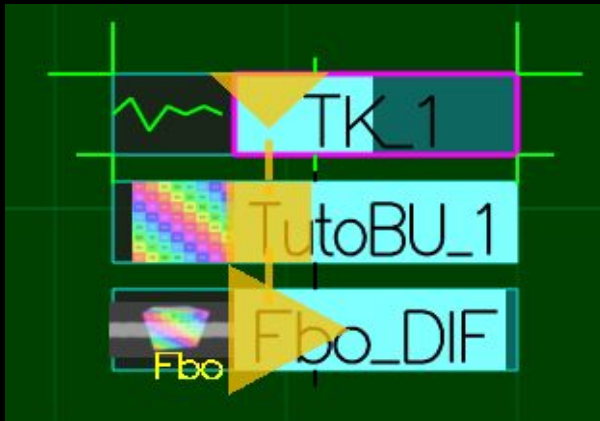
TutoBU\_1 rendered first, then TK\_1



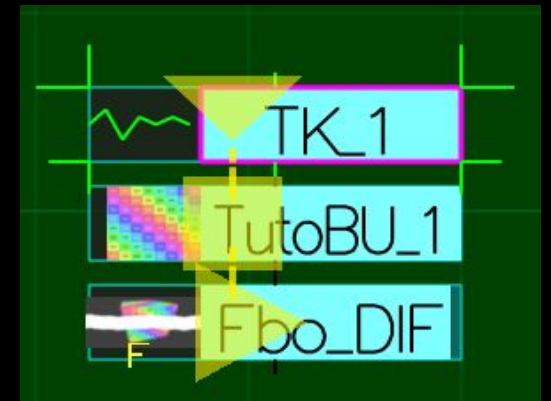
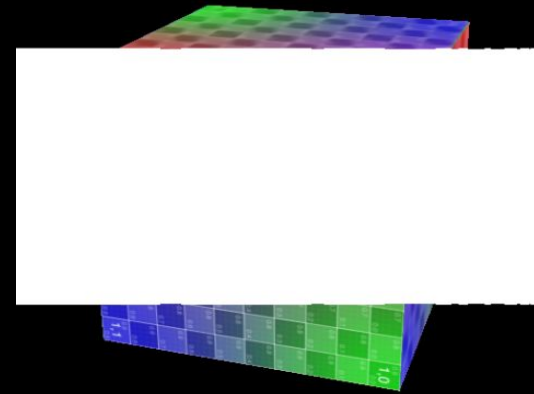
# Render chain: Alpha/Opacity

- mu slider value is the Alpha of the meu's rendering
  - Alpha 1
    - 100% opaque, meu is On
  - Alpha 0.75
    - 75% opaque same as 25% transparent, meu is On
  - Alpha 0
    - 0% opaque same as 100% transparent, meu is Off

TK\_1 is 50% opaque



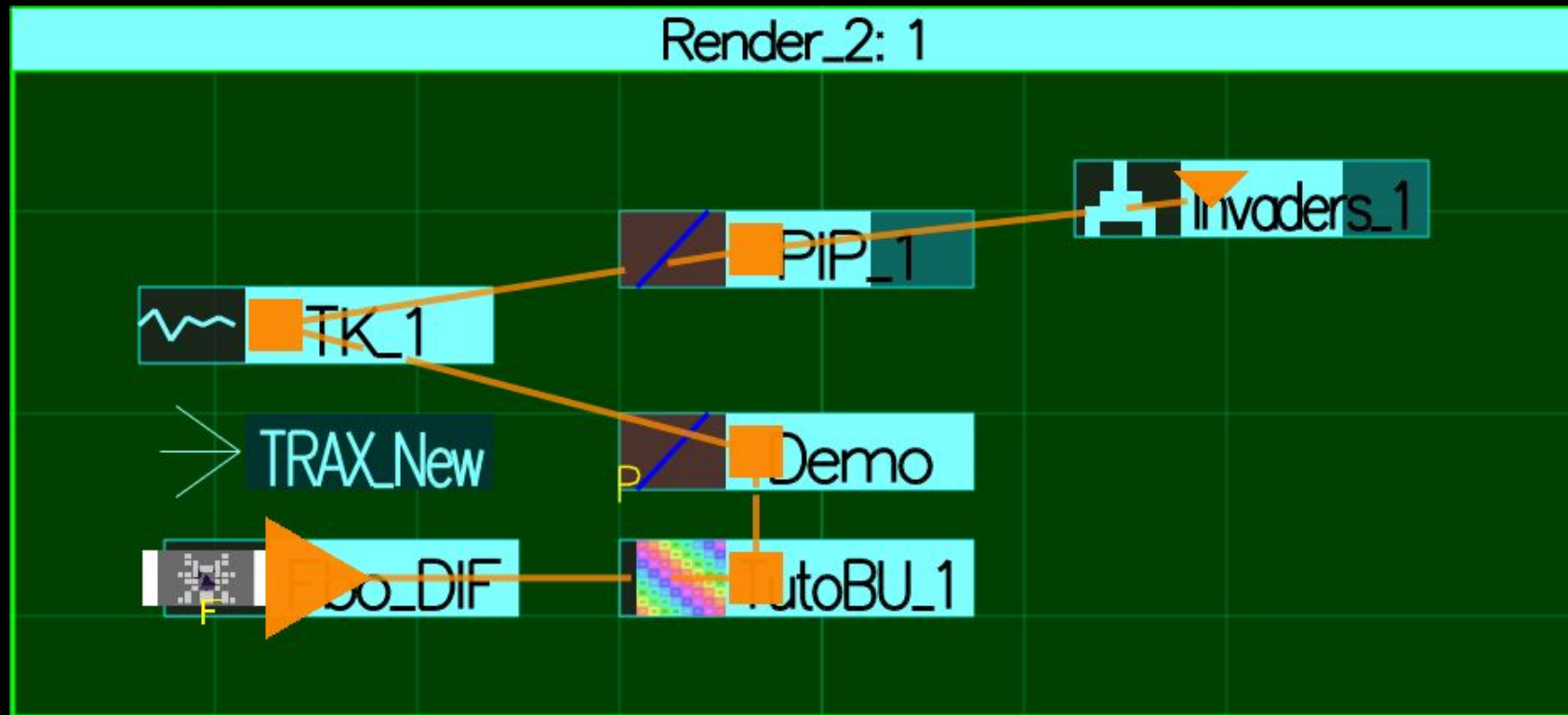
TK\_1 is 100% opaque





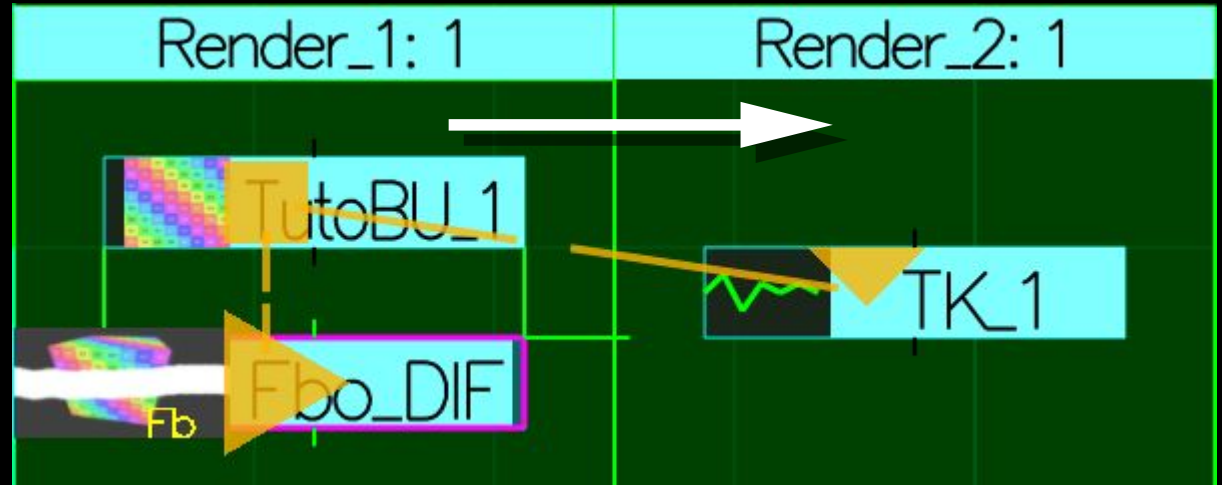
# Render chain: Visualization

- An Orange line **shows** the render chain
- **Just a visualization not a cable !**



# BU\_RECT Order

- A BU\_RECT is a container that holds a group of MUs. It acts as a single control for the group.
- First: Order of the BU\_RECTs  
the number in the name  
to change it  
StarMenu  
Keyboard Ctrl Up / Down
- Second: order inside BU\_RECT  
Bottom to top  
then  
left to right

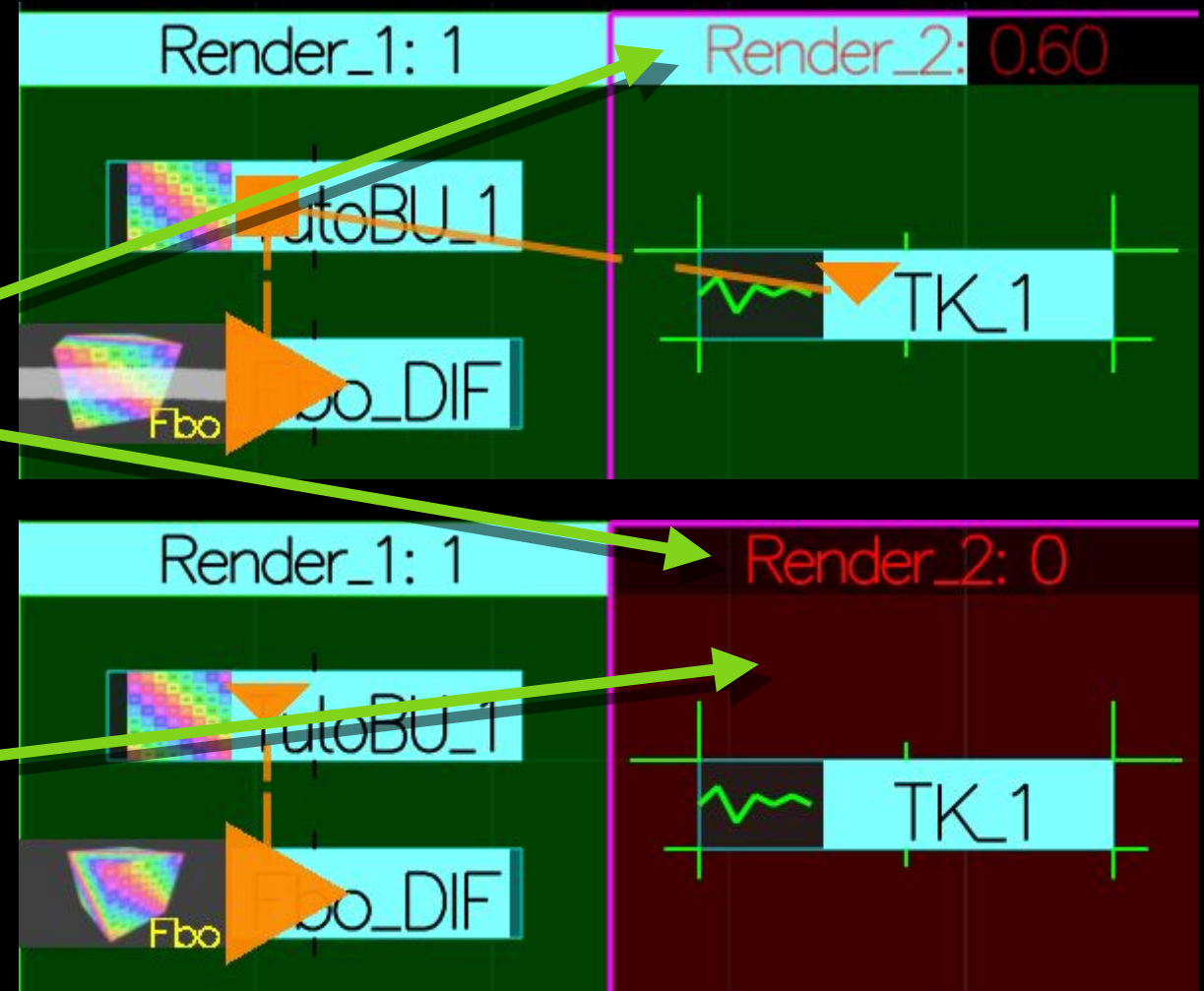


# BU\_RECT Value

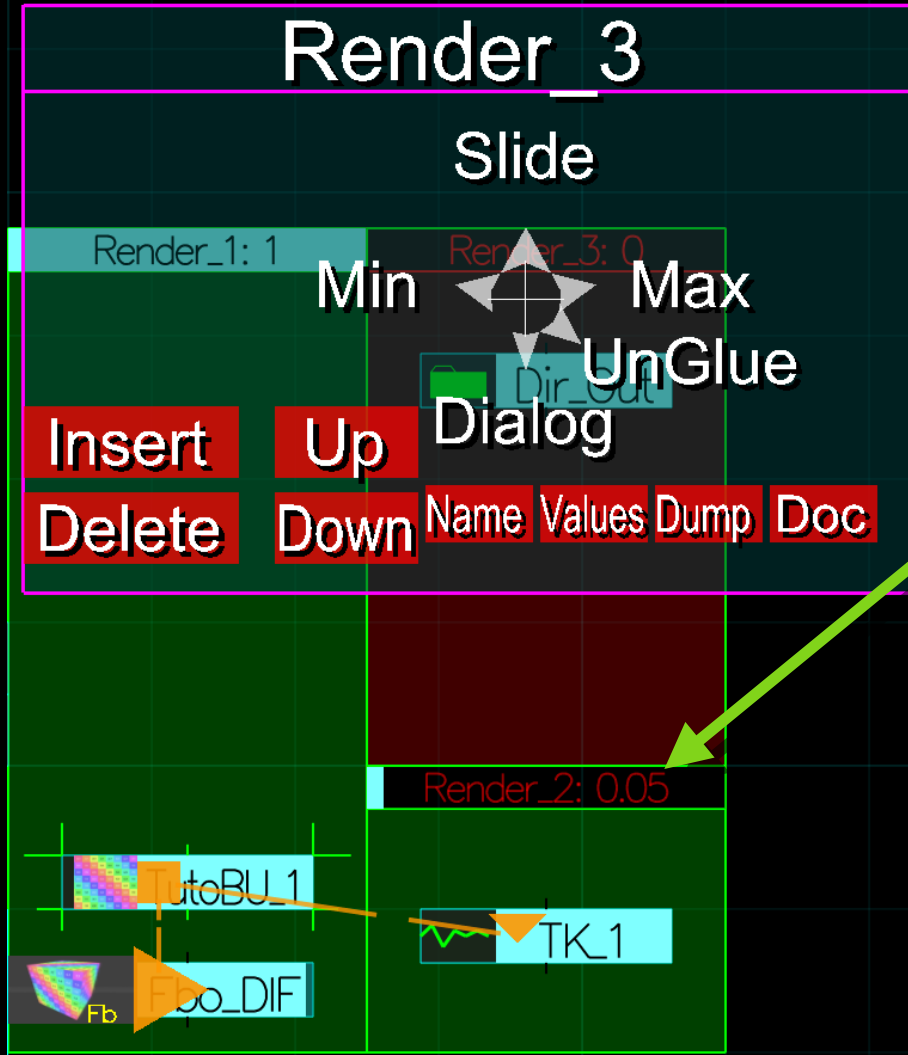
- Control all mu inside:  
it multiply their Alpha.  
Act as a group Alpha.

- Alpha different from 1  
→ red text

- inferior or less then 0  
no mu rendered  
red background



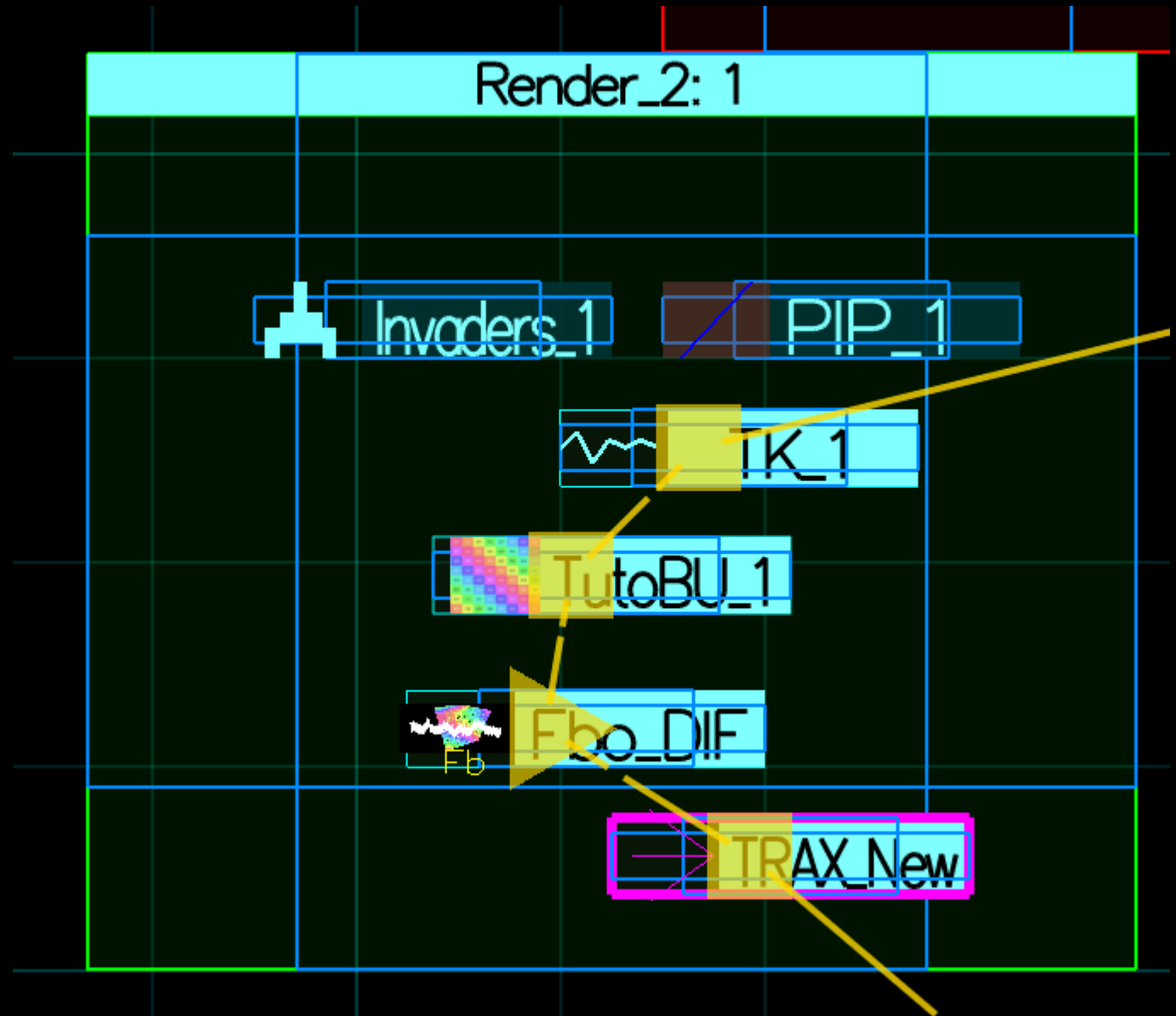
# BU\_RECT StarMenu



- **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





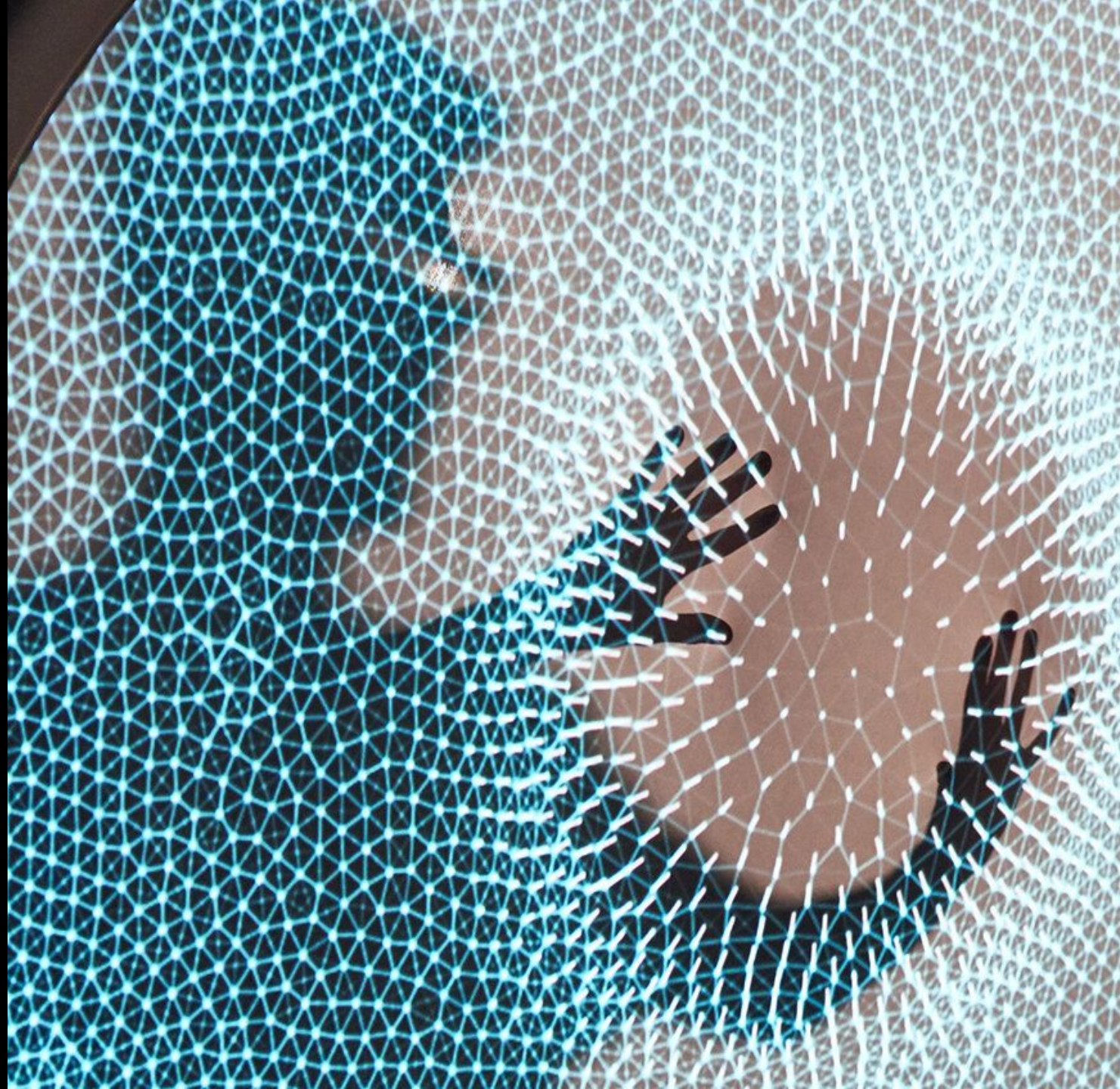
# AAASeed

## 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 controls display of MU with hide attribute to change MU hide attribute**

**MU StarMenu Hide/Unhide Button**

**Shift Ctrl on Icon**

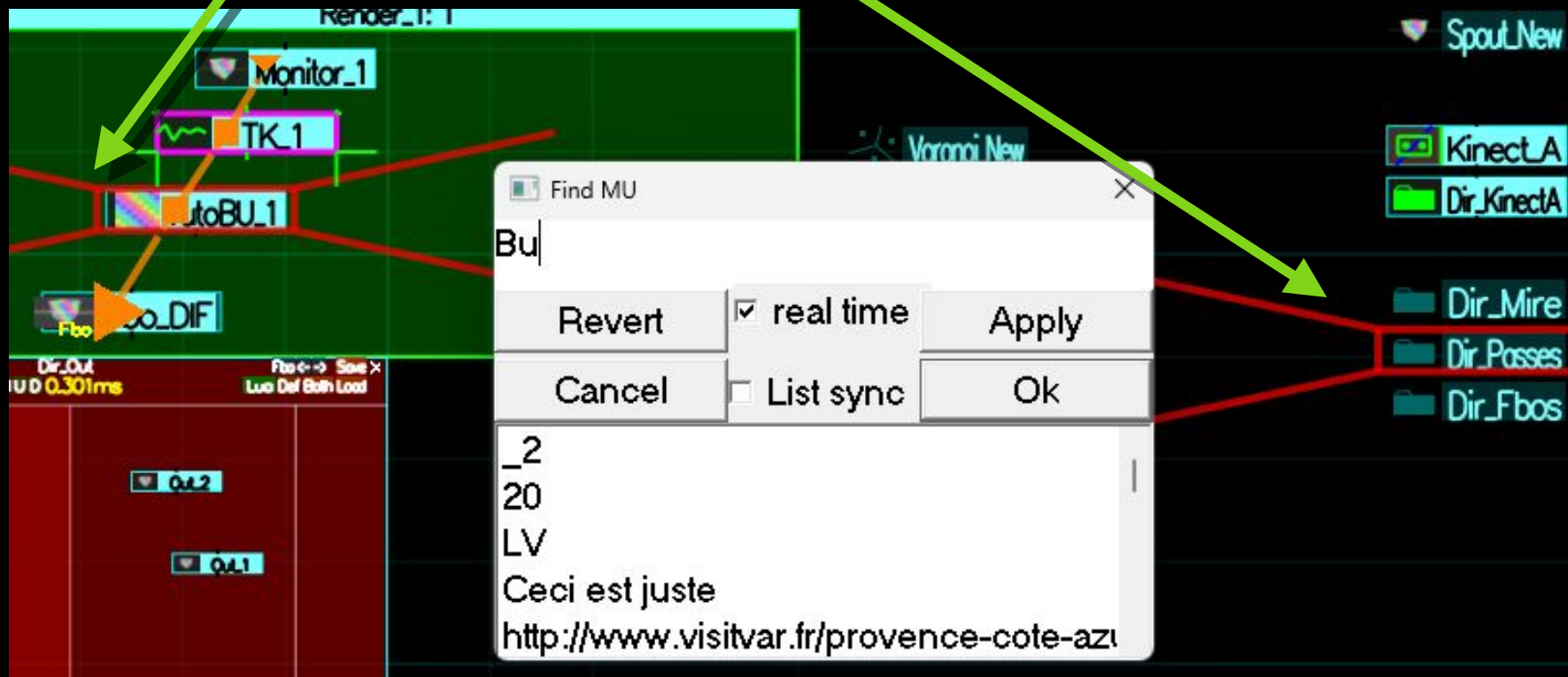
**Hide On → red background on MU**



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

# Finding MU

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



# Moving MU (reminder)

- Move Resize as always

Hold Alt

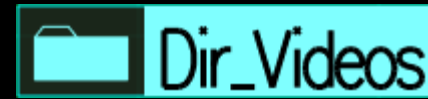
Drag in Zone

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,
  - while 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 MEU\_DIR
  - They open in Ui5 so the MEU\_DIR Ui stay open
- 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 folder AAA\_MEU to store its MEUs





# MEU name: Type\_Instance

- 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 same lua script: one lua script by meu Type**.
- **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**

for example you will find the preset file there

# MEU Instance operation

- StarMenu

- Rename

- change the name of the Instance
    - Type part can't be changed

- Destroy

- delete permanently the instance
    - with Confirmation Dialog

- Instantiate

- Duplicate, create a copy with another name
    - Type Name can't be changed



Can't instantiate MEU of type  
TK  
with a different type name  
Ts

Really want to destroy  
TK\_1  
No Undo here for now

No Destroy

MEU|TK\_1 Can't rename MEU of type  
tk  
with a different type name  
tee

- 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
- The state of the BU making the interface of the MEU in file preset\_0.plua  
other preset are saved or deleted at the time the action is performed 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 proved very useful  
you can experiment and save only what you want  
eventually use Load on some MEU to reset them to their saved 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

Save All

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



## Create New APP: Duplicate folder

- APP are stored in a folder
- Example: duplicate APP\_Garden\_Base

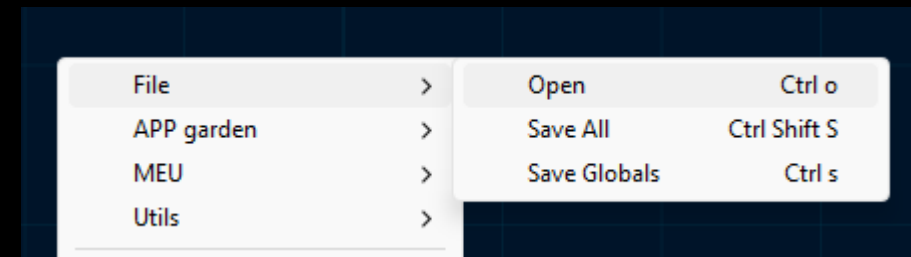
Copy AAAAPPs\Tuto\APP\_Garden\_Base

Into APPs\_Guest

Rename APP\_Garden\_Base

Open AAASeed

Open renamed APP





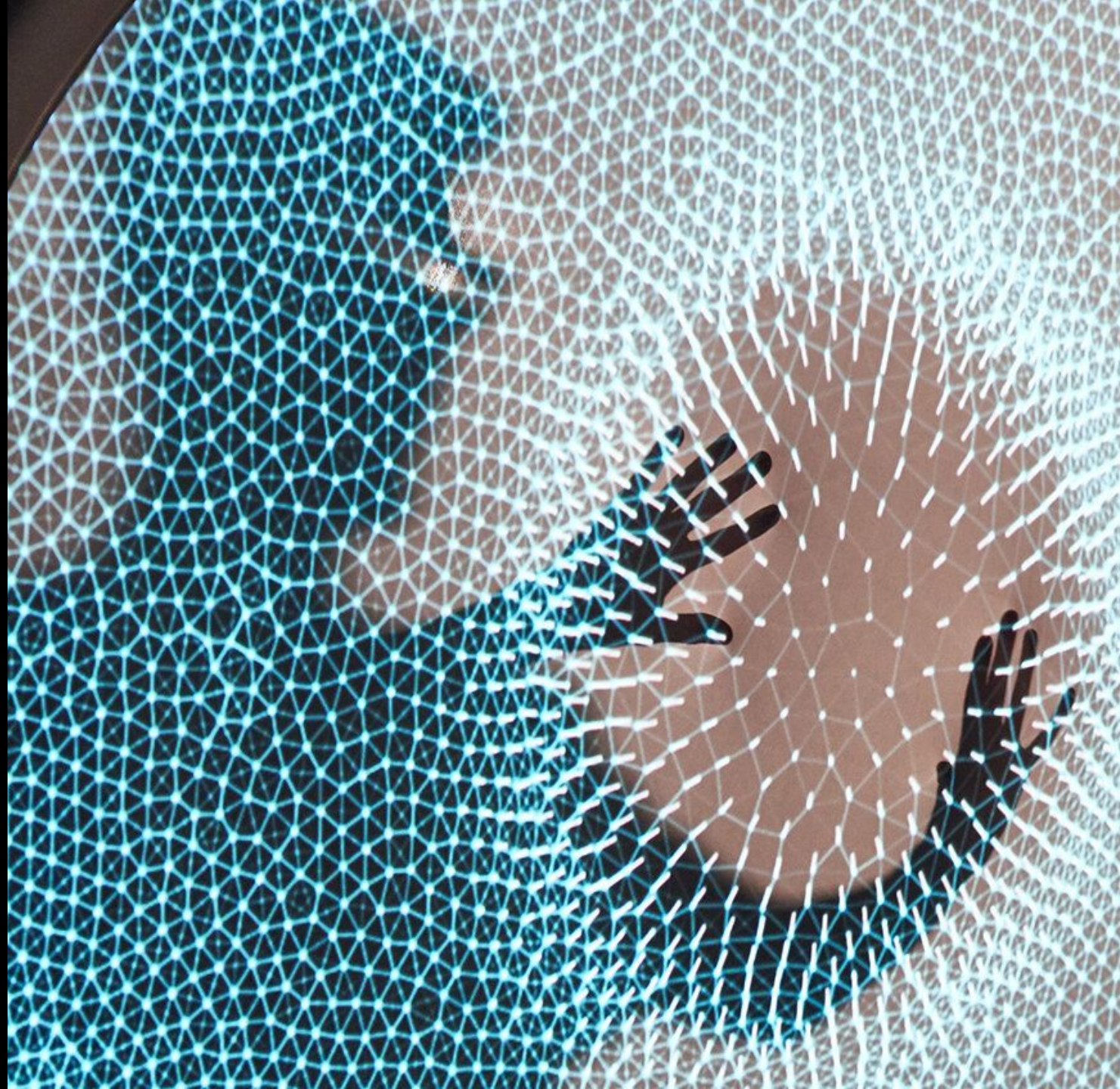
# AAASeed

## An introduction

### Part 12:

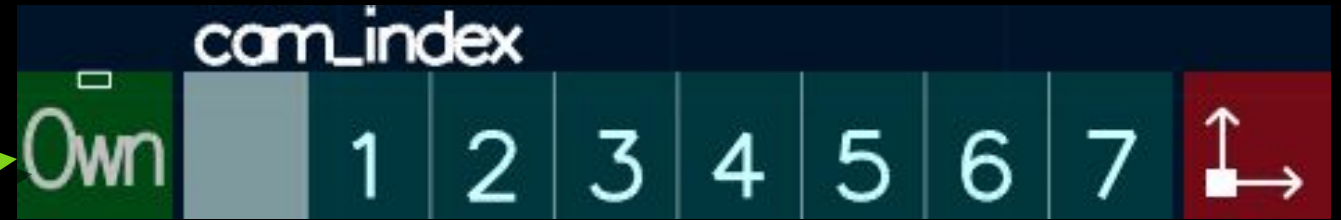
### Cameras & Axe

- MEU's Cameras
- Camera Selector
- Axes Units
- Rotation
- Camera Edit
- Cameras Params



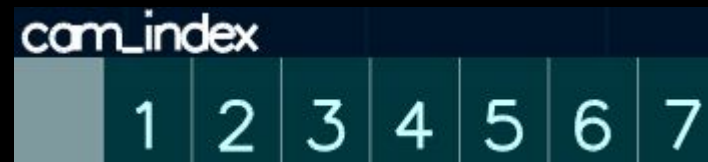


# MEU's Cameras



- **No / Once / Own**
  - **No** → **No camera defined: use current camera**  
the last one defined in the render chain (**Own** see below)
  - **Once** → **use camera only for this MEU**  
then go back to current camera
  - **Own** → **Own the camera**  
it becomes the current camera  
for this Meu and the next Meus in the render chain

- Camera selector

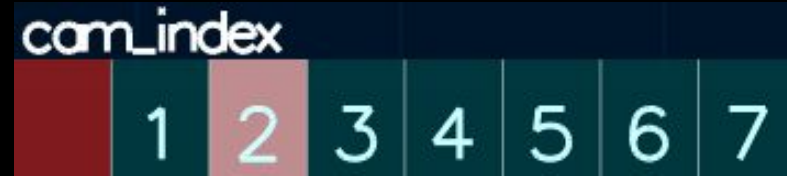


- Button Axe  
Toggle Axes



- Reminder: StarMenu to see the BUs name

# Camera Selector

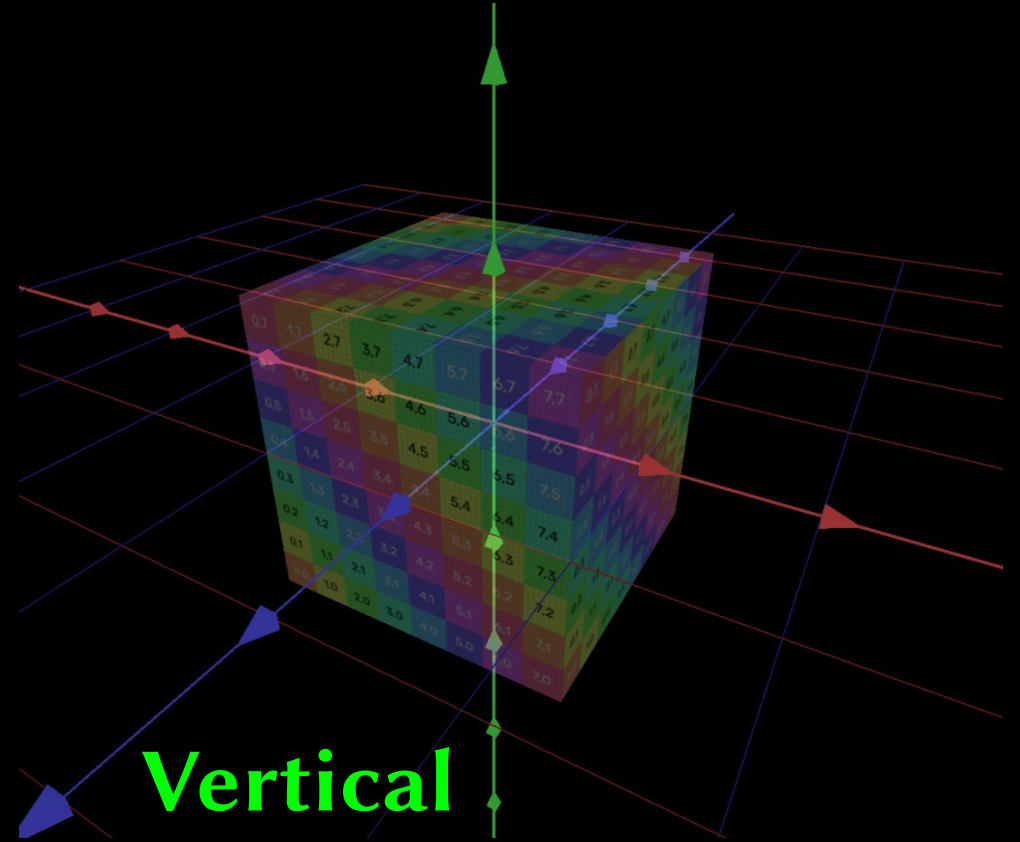


- 8 cameras by default, 16 sometimes
- Click → Select camera
- Ctrl Click → Copy the last camera clicked
- Ctrl C/Ctrl V → Copy / Paste
- Double Click → Edit in flatland
- Shift Click → Flip Lock / Unlock

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**
- **Z|Blue**        **back to front**
- **No unit: Mathematics not physics**  
Back of the pyramid on axes aligned with the unit



# Rotation

- Unit is Turn / Revolution

No radians, degrees or gradians.

Canonical aspect

[0,1] map full circle

1 → full turn

0.5 → half turn

0.25 → quarter turn (counter clockwise)

-0.25 → quarter turn reverse (clockwise)

- Rotations use trigonometric direction (Counter Clockwise)

# Camera Edit

- **Middle Click → Start Editing**

BU\_CAM will manifest if camera edit is locked

- BU\_CAM itself can be 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**
- **end with Left Click** → **accept edit**
- **end with Right Click** → **cancel edit: go back to position at beginning of edit**



# Camera Params

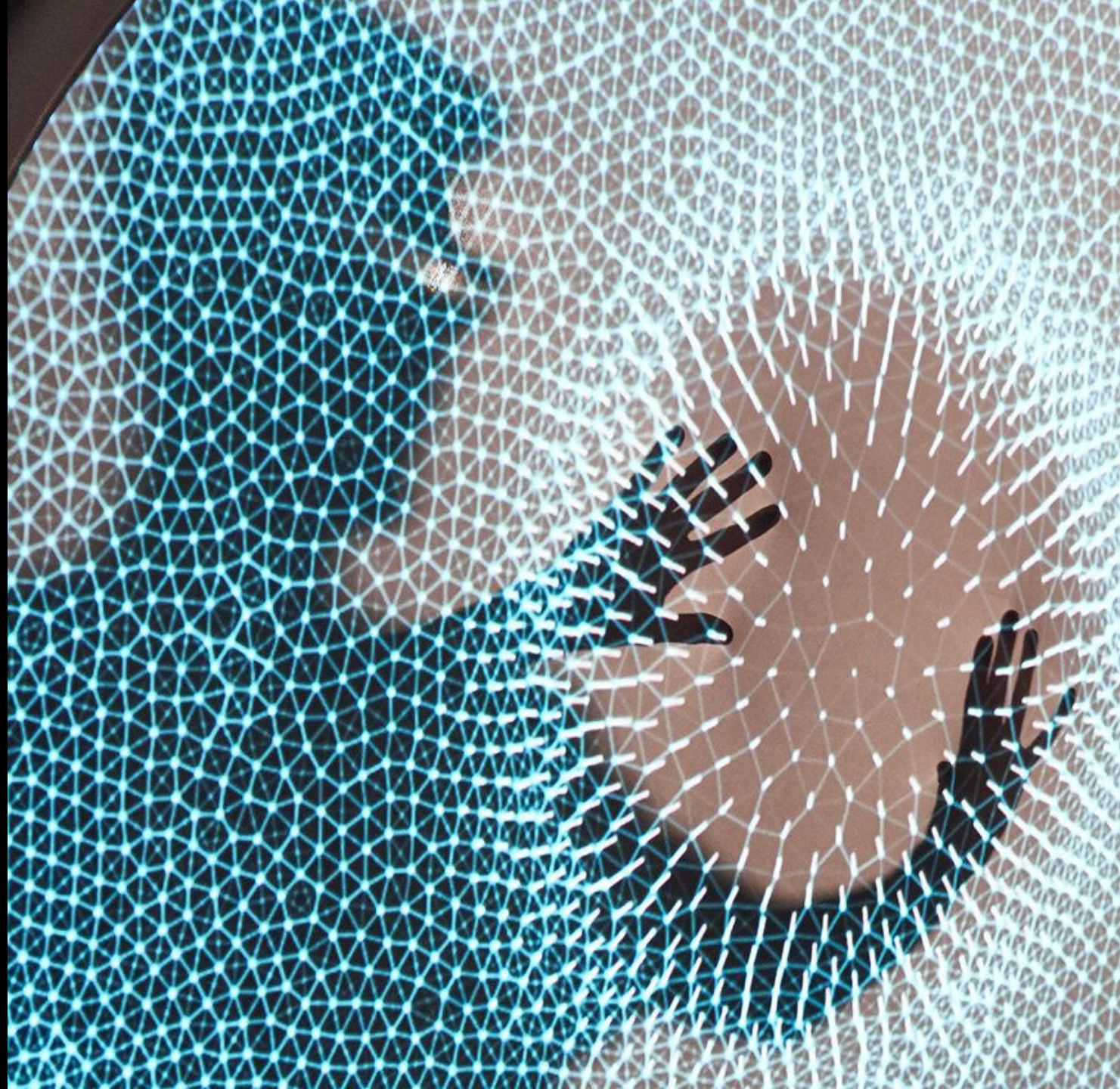
- To be added

# AAASeed

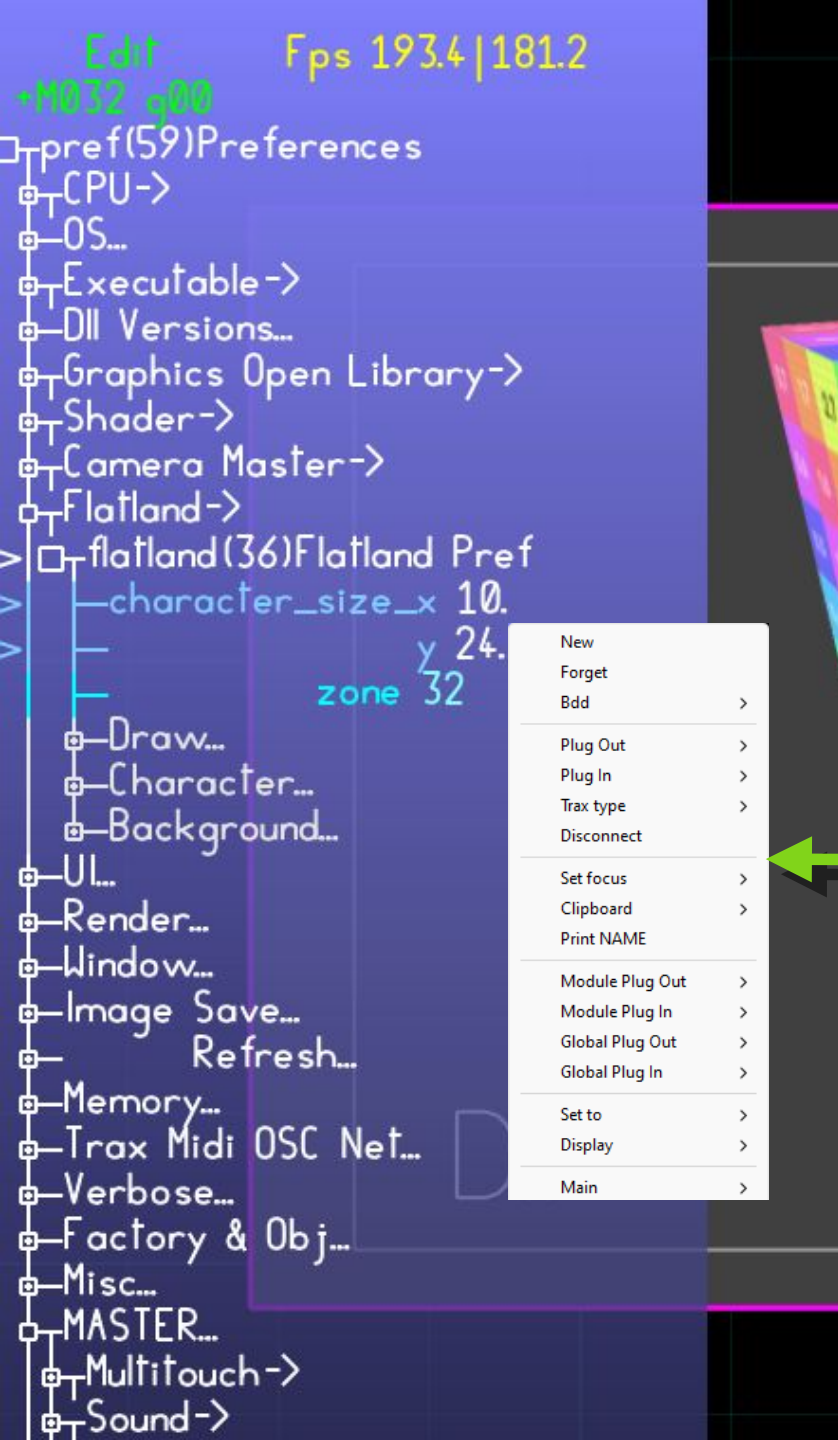
## An introduction

### Part 13: Flatland and Param

- Flatland 101 reminder
- Param
- Navigate
- Focus
- Param Edit
  - **Keyboard**
  - **Mouse**

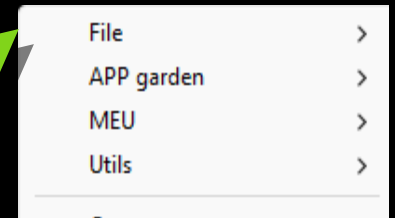


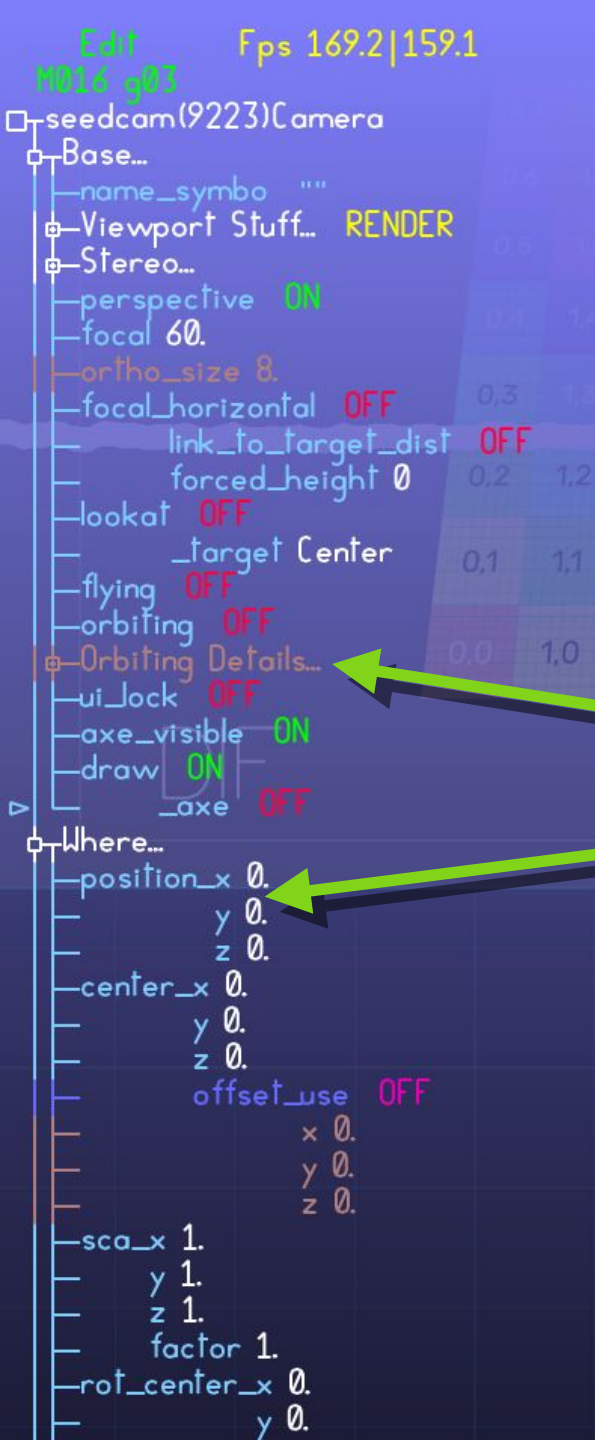




# Flatland 101 reminder

- **Original AAASeed Ui** from previous century  
Name from « **Snow Crash** » book by Neil Stephenson  
where the **C++** can be seen and used  
made of **c\_obj\_ui** and **param** (for parameter)  
Tree Made of params  
developer will say render graph
- **Tab** → switch flatland On/Off
- **Right Mouse Button** → **Menu**  
in Flatland → **Param menu**  
out Flatland → **Main menu**  
**GaBuZoMeu menu**  
**Beginning of Main menu**
- **Flatland** have precedence for events and keyboard
- Needs to be off to use keys with **GaBuZoMeu**





# 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:**

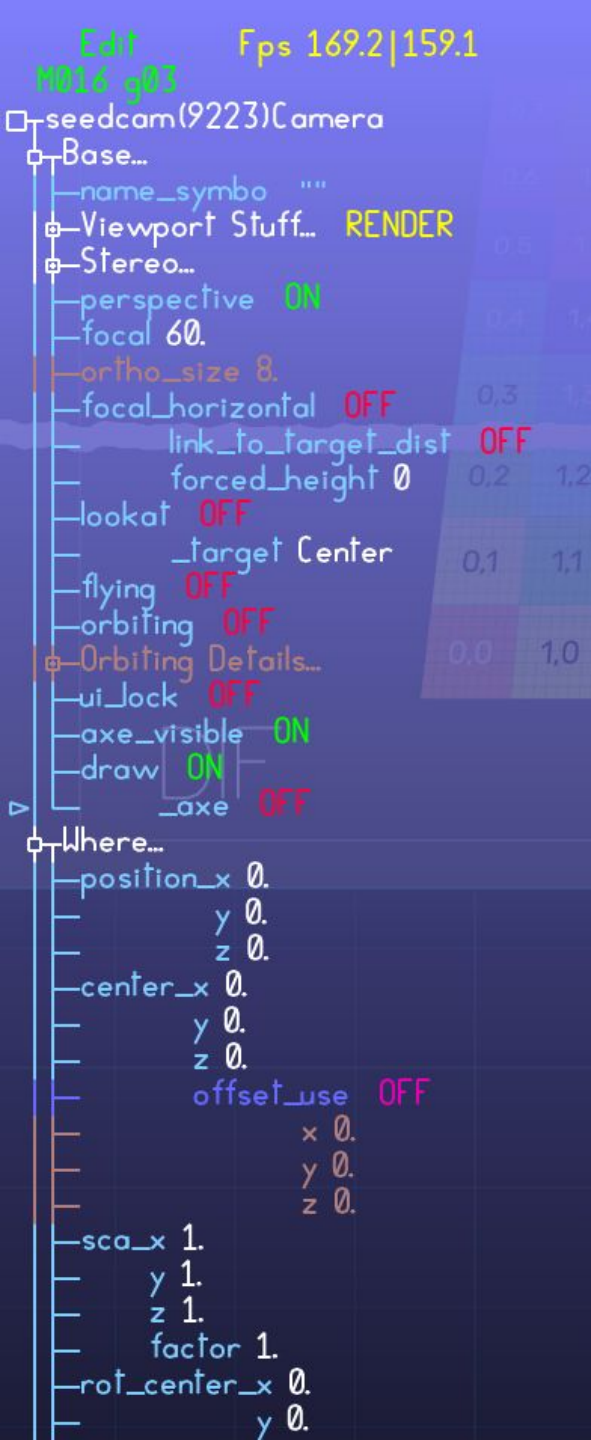
– **Big triangle pointing left just before name**

parameter is generated by C obj it can't be edited

– **Small triangle pointing right at beginning**

reference is defined

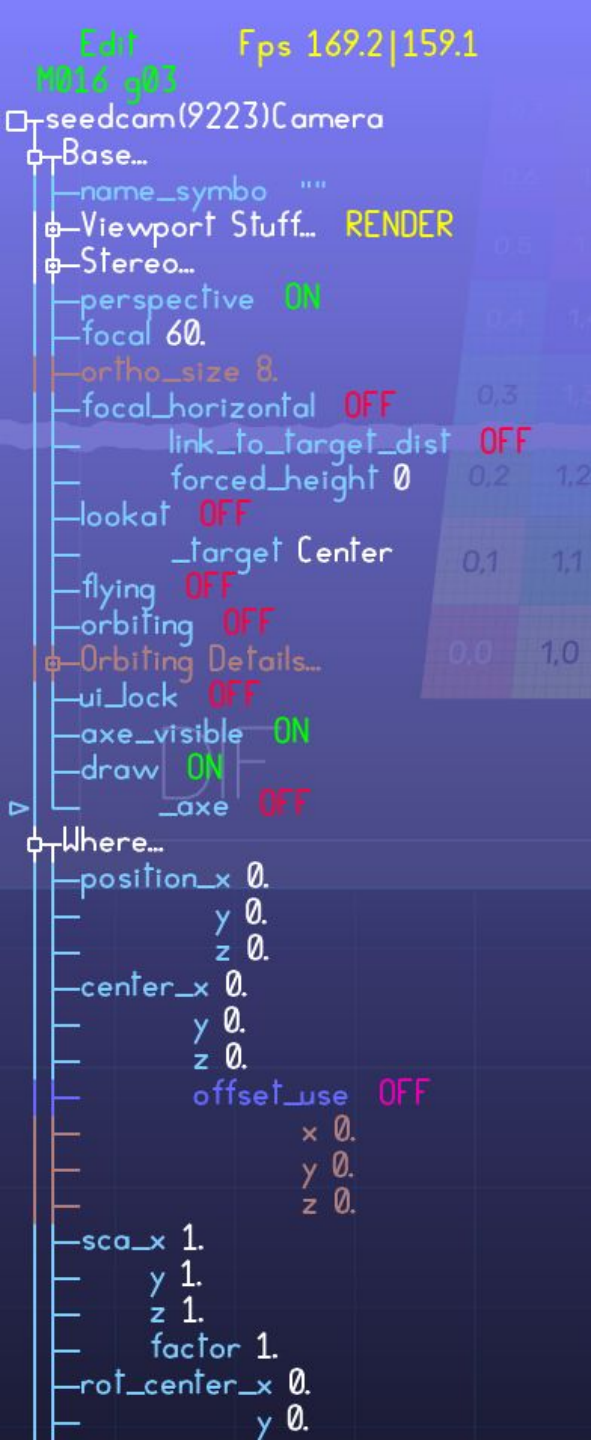




# Navigate Params

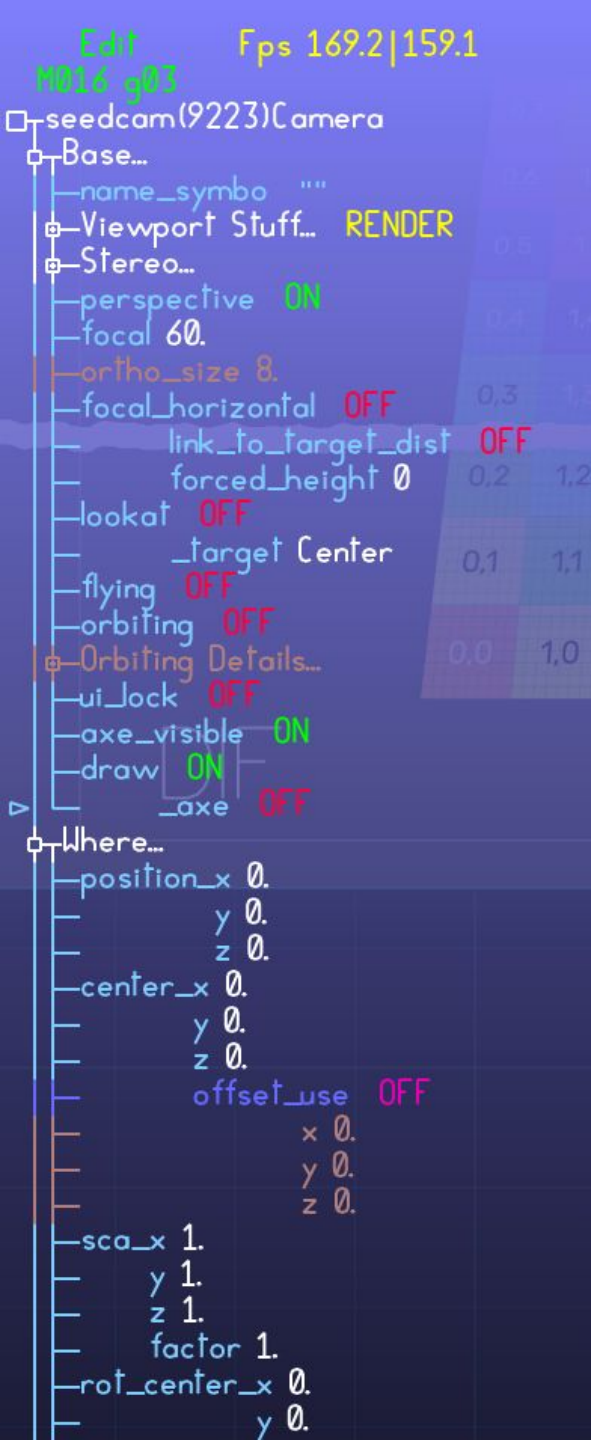
- mouse wheel  
→ Scroll
- Click  
Left  
→ Set current param  
→ Open / Close if possible  
Right  
→ Menu  
depends on position
- Arrow key Up/Down  
→ Change current param Up/Down
- Space  
→ Open/Close if possible





# Flatland Focus

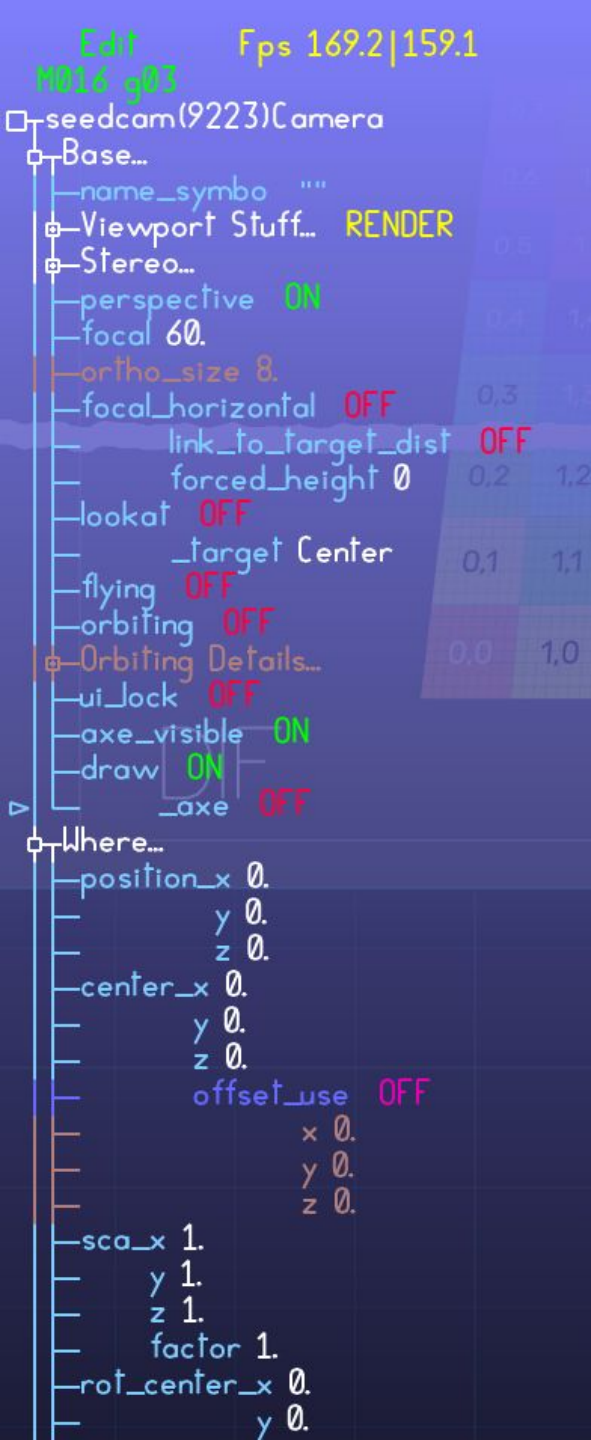
- focus means the object shown in Flatland the root of the tree
- Keys set focus on
  - F4 → current camera
  - F10 → Preferences
  - Double F10 → Start preferences
  - Triple F10 or 0 → Application
  - 2 → current c\_module
  - 3 → current c\_layers
  - 4 → current c\_layer
  - Triple n → (n)etwork preferences
- Arrow key Left / Right → Navigate in Focus history



# Param Edit: Keyboard

Similar to BU change value

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



# Param Edit: Mouse

- **Boolean**  
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**  
→ trigger Dialog or Action if available for param
- **Developer note**  
Alt Middle Click → flip debug view for param



# 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

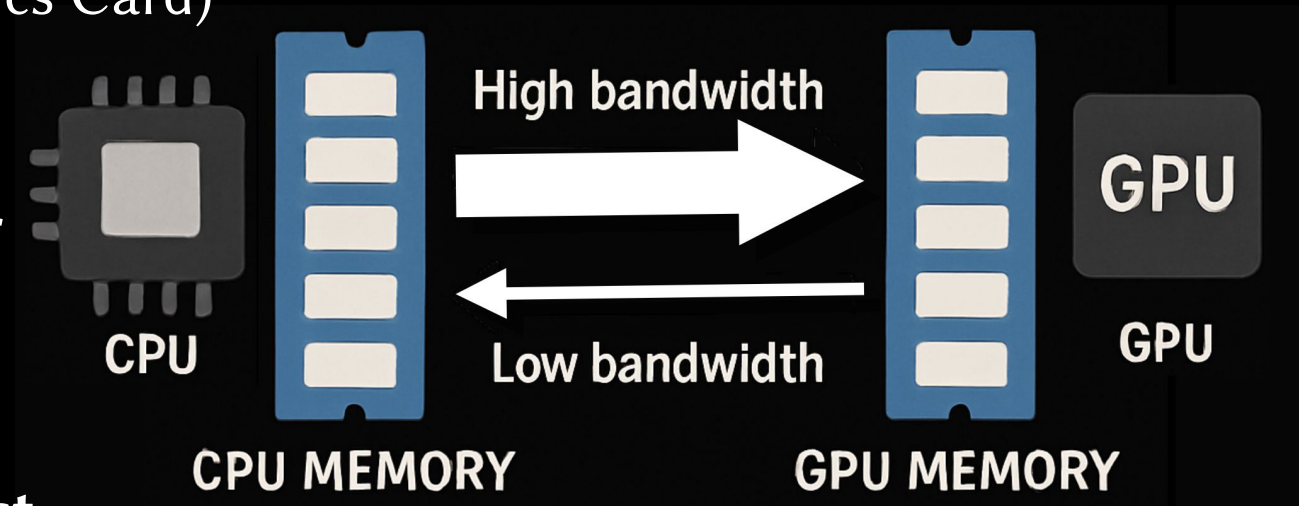
- GPU to CPU transfer is slower**

- can cost a lot

- developer note

- it stall the pipeline

- GPU memory can be really fast**

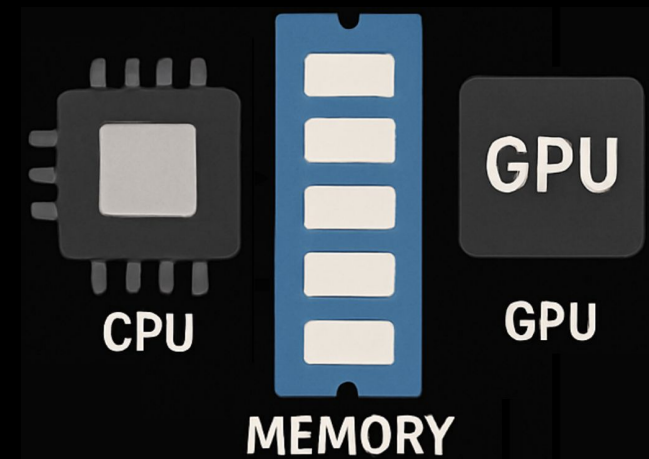


- **Unified Memory (Integrated Graphics)**

- No memory transfer**

- Memory is slower**

- than dedicated GPU memory



# Texture channels

See BU\_MONITOR part

– 1 or R for Red  
showed as greyscale

1920 x 1080 Red 8



– 2 or RG for Red + Green  
showed as yellow  
no Blue

1920 x 1080 RG 8



– 3 or RGB for Red + Green + Blue  
full color

1920 x 1080 RGB 8



– 4 or RGBA for RGB + Alpha  
like RGB but with transparency/Opacity  
0 mean transparent, 1 Opaque

1920 x 1080 RGBA 8



# 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]  
 $\text{flat bind} = \text{bank} * 32 + \text{bind}$

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

# BANK

Slow blinking redish  
Blue diagonal

## Bank selector

bank contains 32 binds

# Bind in bank

## [0,31]

Name

**Flat bind**  
**[0,65535]**

## Each bind a Monitor:

## Click Double or Space

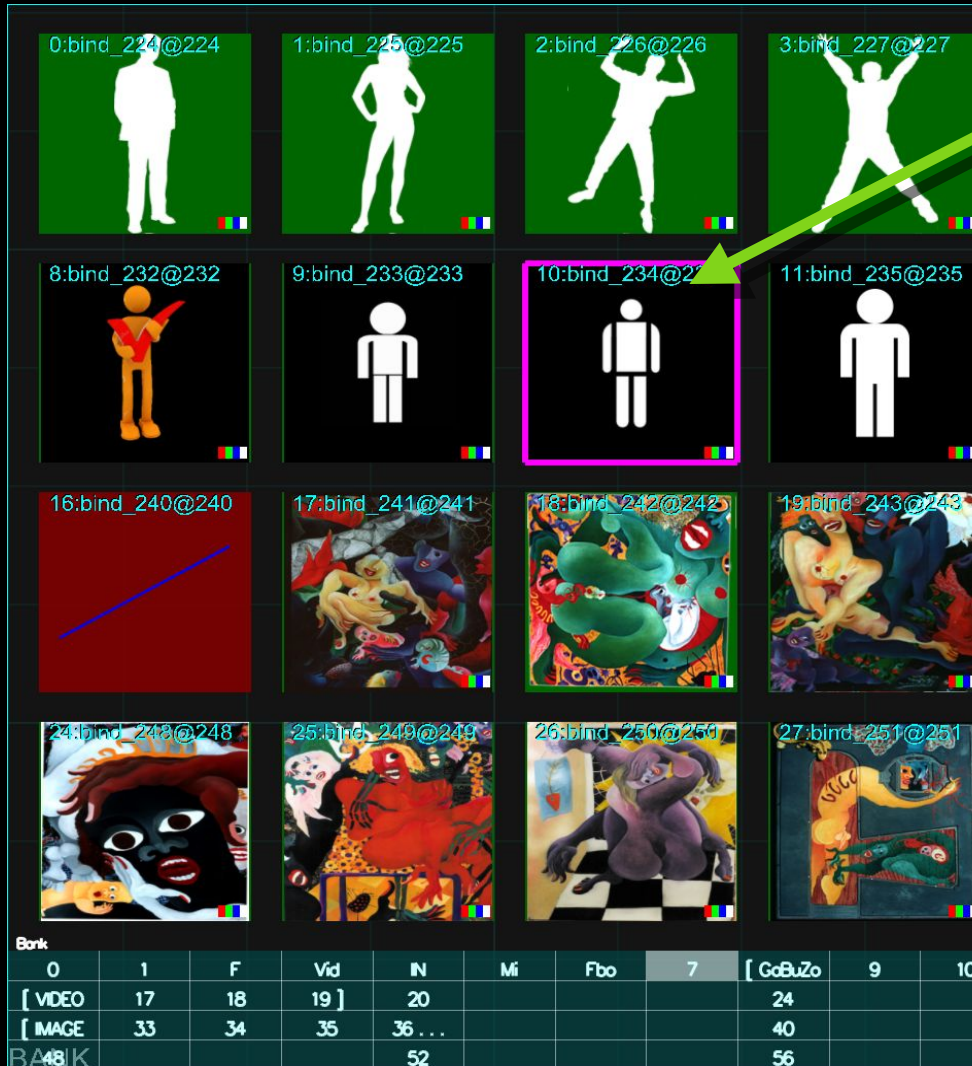
## Open monitor

## StarMenu Extended

**bank offset**  
0 to 1984



# BU\_BANK Edit



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

Ctrl Arrow

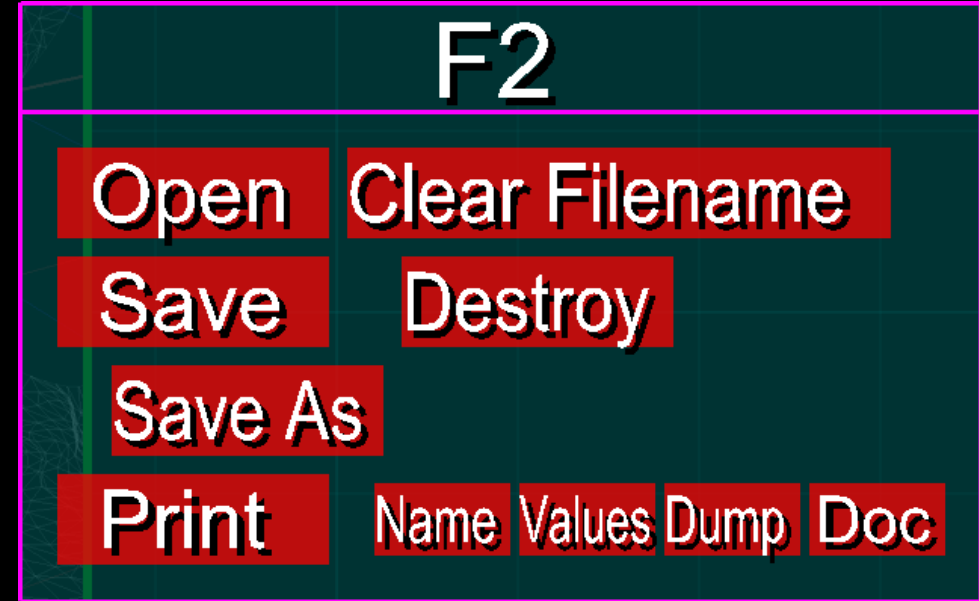
Swap texture and filename with the bind in the arrow direction  
Stay in the current bank

Ctrl C/Ctrl V

Copy/Paste texture and filename  
Can be done across bank

# BU\_MONITOR StarMenu Reminder

- Use **Quick drag-out** like any BU where it doesn't appear directly
- **Open**
  - Load image(s) via a file dialog  
at the current bind  
later on this (Part 14: Texture)  
can choose several files  
loaded in successive binds
  - Set the filename for this bind
- **Save**
  - Save the Image to Media/AAASnapshots  
How it is saved is 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 is in Flatland Prefs/MASTER/Print
- **Clear Filename**
  - Clear the the filename for this bind
- **Destroy**
  - Destroy the texture for this bind  
Clear the filename too: no more pixels or filename left



# BU\_MONITOR BU\_BANK StarMenu

Same StarMenu with more commands



- **Dec**  
(Dec)rement the texture index  
in fact swap with the previous one  
Same as Ctrl Left
- **Inc**  
(Inc)rement the texture index  
in fact swap with the next one  
Same as Ctrl Right



# AAASeed

## An introduction

### Part 15:

### Texture More

- An apology
- BU\_BANK and BU\_TEXTURE
- Texture Name
  - **More**
- BU\_TEXTURE
  - **Bank button**
  - **Bind Slider**
  - **Shortcut BU**





# Disclaimer on current texture architecture

- It is a problem  
Mâa apologize deeply
- We now it, it is tricky  
Bank fixed size  
No User Bank

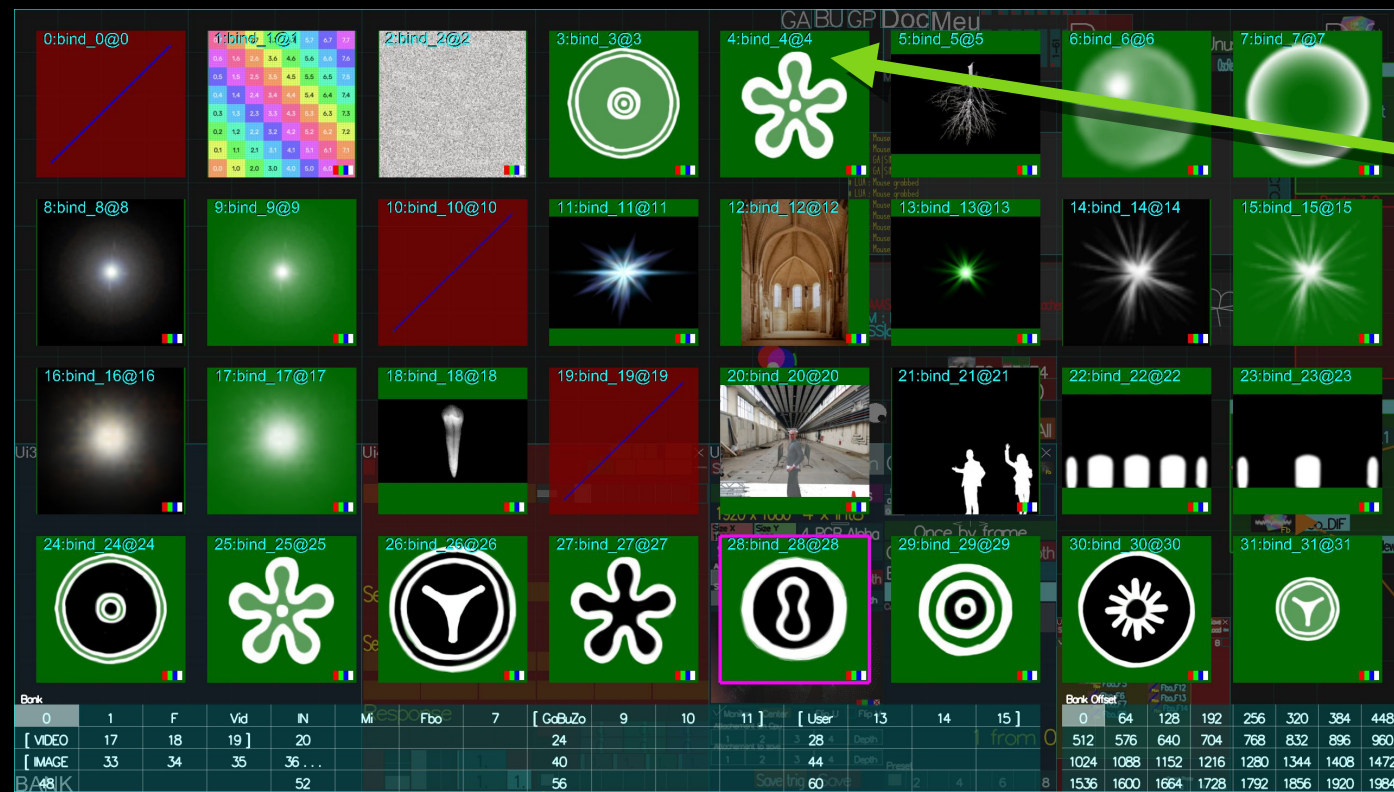
...



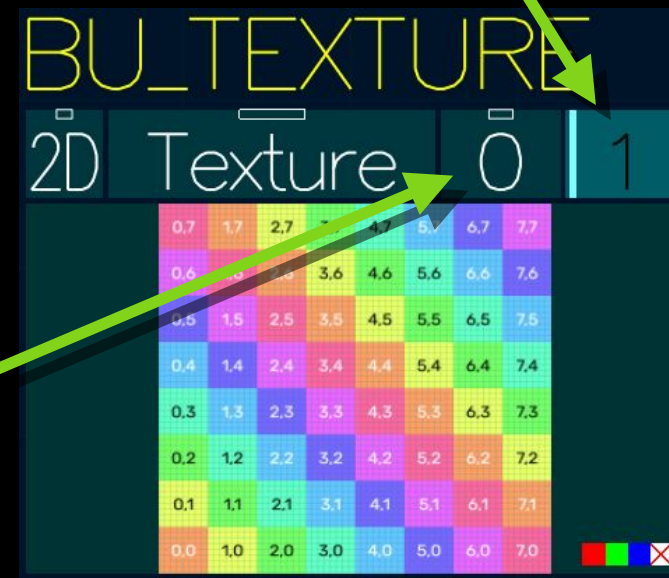
- We could not change it before the end of the ArtCast4d project  
There will be MEU Bank to define and edit them  
Future Banks will have names (not only number) and each their own bind size
- We have to live with it for a while (2025)  
This is stable and solid.  
The new texture architecture will probably keep the current structure at the bottom and build on top of it.



# BU\_BANK and BU\_TEXTURE



bind



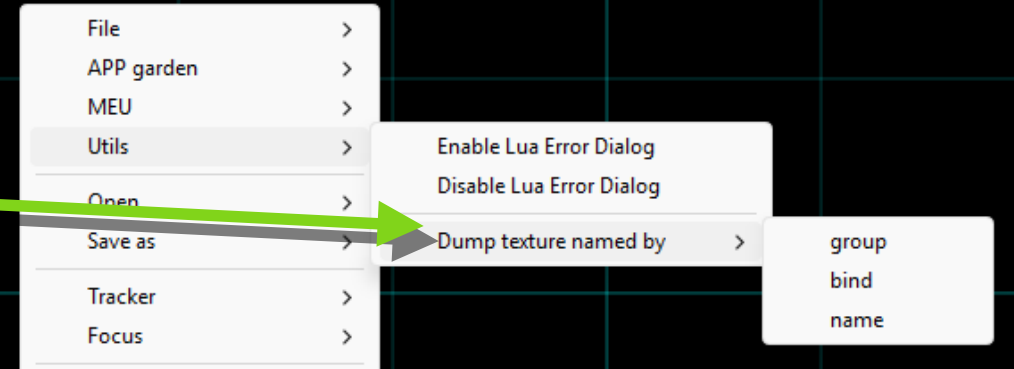
bank

# 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  
name are made of alphanumeric character and dot (.)
- **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 of these 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

**Bank button** to choose the bank

Click open a selector

Texture_bank																	×
0		F	Vid	IN	Mi	Fbo		[ CoBuZo	9	10	11 ]	[ User	13	14	15 ]		
VIDEO	17	18	19 ]	20				24				28				31	
[ IMAGE	33	34	35 36 ...					40				44				47	
48				52				56				60				63	

**Double Click in these selectors**  
triggers a dialog to load texture in current bind

Texture_bind								×
0				4			7	
8				12			15	
16				20			23	
24				28			31	

**Dimension**  
(Optional)  
up to the MEU

Texture dim					×
no	1D	2D	3D		

BU\_TEXTURE

2D

Texture

0

1

0.7	1.7	2.7	3.7	4.7	5.7	6.7	7.7
0.6	1.6	2.6	3.6	4.6	5.6	6.6	7.6
0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5
0.4	1.4	2.4	3.4	4.4	5.4	6.4	7.4
0.3	1.3	2.3	3.3	4.3	5.3	6.3	7.3
0.2	1.2	2.2	3.2	4.2	5.2	6.2	7.2
0.1	1.1	2.1	3.1	4.1	5.1	6.1	7.1
0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0

**Bind slider** to choose the bind  
click open a selector and starMenu

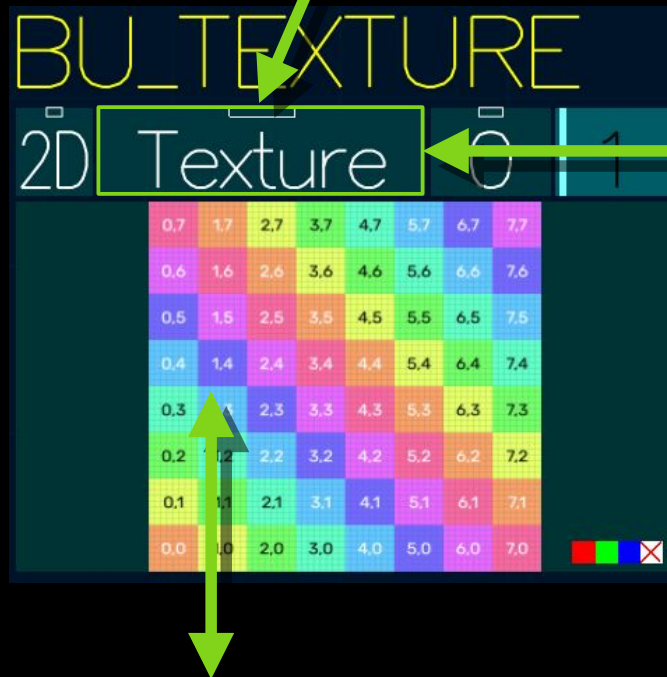


- **BU\_TEXTURE Shortcut BU**

- A shortcut BU simplify the manipulation

Click open a window with the most used texture names

StarMenu provide also a fast way to navigate and select one texture



Video

Texture				Fbo				Input				Mire				Monitor				✕
A	B	C	D	F1	2	3	4	In1	2	3	4	Mi1	2	3	4	M1	2	3	4	
E	F	G	H	F5	6	7	8	In5	6	7	8	Mi5	6	7	8	M5	6	7	8	
I	J	K	L	F9	10	11	12	In9	10	11	12	Mi9	10	11	12	M9	10	11	12	
Dif	Pre	Dif	R Viz	F13	14	15	16	In13	14	15	16	Mi13	14	15	16	M13	14	15	16	
Albe	Nor	Spe	Emis	Z	Buf	Light	Caus	Fog	Dof	Trans	God	Tone								UV
do	mal	cular	sive							parent	Rays	Map	Fx00							Map

- Monitor can be used here too (Click zones, StarMenu, Keyboard)



# AAASeed

## An introduction

### Part 16:

### Fbo

- Frame Buffer
  - Canvas
- Fbo/Frame buffer object
  - **Attachments**
- Fbo render chain
  - **No Fbo**
  - **MultiPass example**
- MEU Fbo name
  - **DIF name**
- MEU Fbo Ui
  - **Main tab**
  - **Copy tab**



# 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 Buffer or Fbo are specific MEU
- A Fbo regroups textures (images) of the same size called **attachments**.
  - Color attachments
    - we use 1 most of the time, but AAASeed deal with up to 8 if needed.
  - Depth attachment (used to do z-Buffering)
- Fbo are needed to draw into textures.
- AAASeed use a MEU\_Fbo

Special MU display

Icon changing size

Fbo yellow letters animated

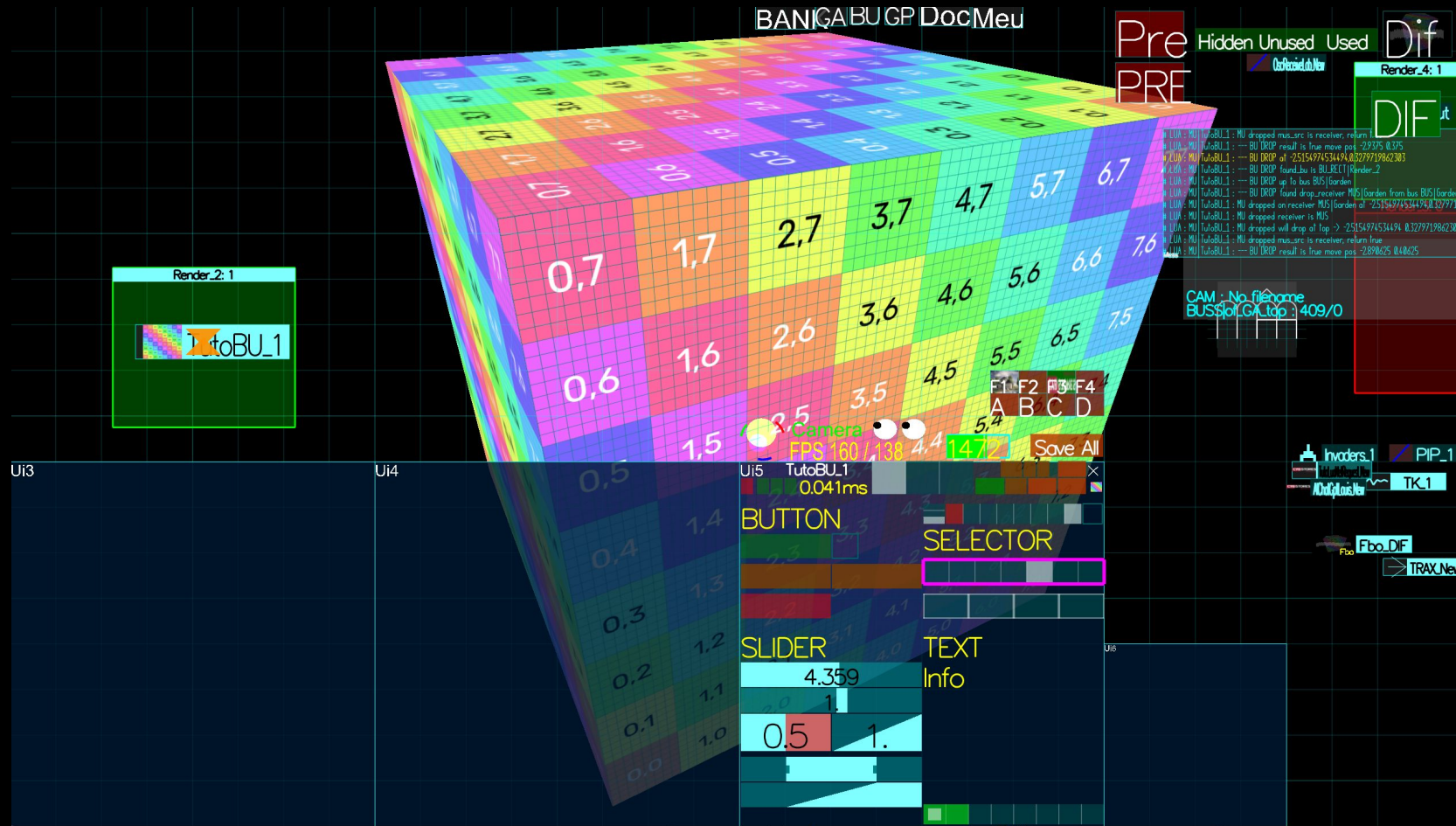




# Meu\_Fbo and render chain: No Fbo

At the beginning of the render chain there is no Fbo defined yet

So AAASeed draw directly to the back of the window, under the Ui.





# Meu\_Fbo and render chain: Multi pass example

As soon a MEU\_Fbo is executed in the render chain

All the following Meu in the render chain use it  
until a new Meu\_Fbo is executed and changing the  
current Fbo

Here (bottom to top)

1- Meu Fbo\_F1 is executed

it defines a 1024 x1024 texture 1 channel 8 bits

it erases itself with grey and become the current Fbo

2- Meu Invaders is drawn in the current Fbo: F1

3- Meu TK\_1 is drawn in the current Fbo: F1

4- Meu Fbo\_DIF is executed

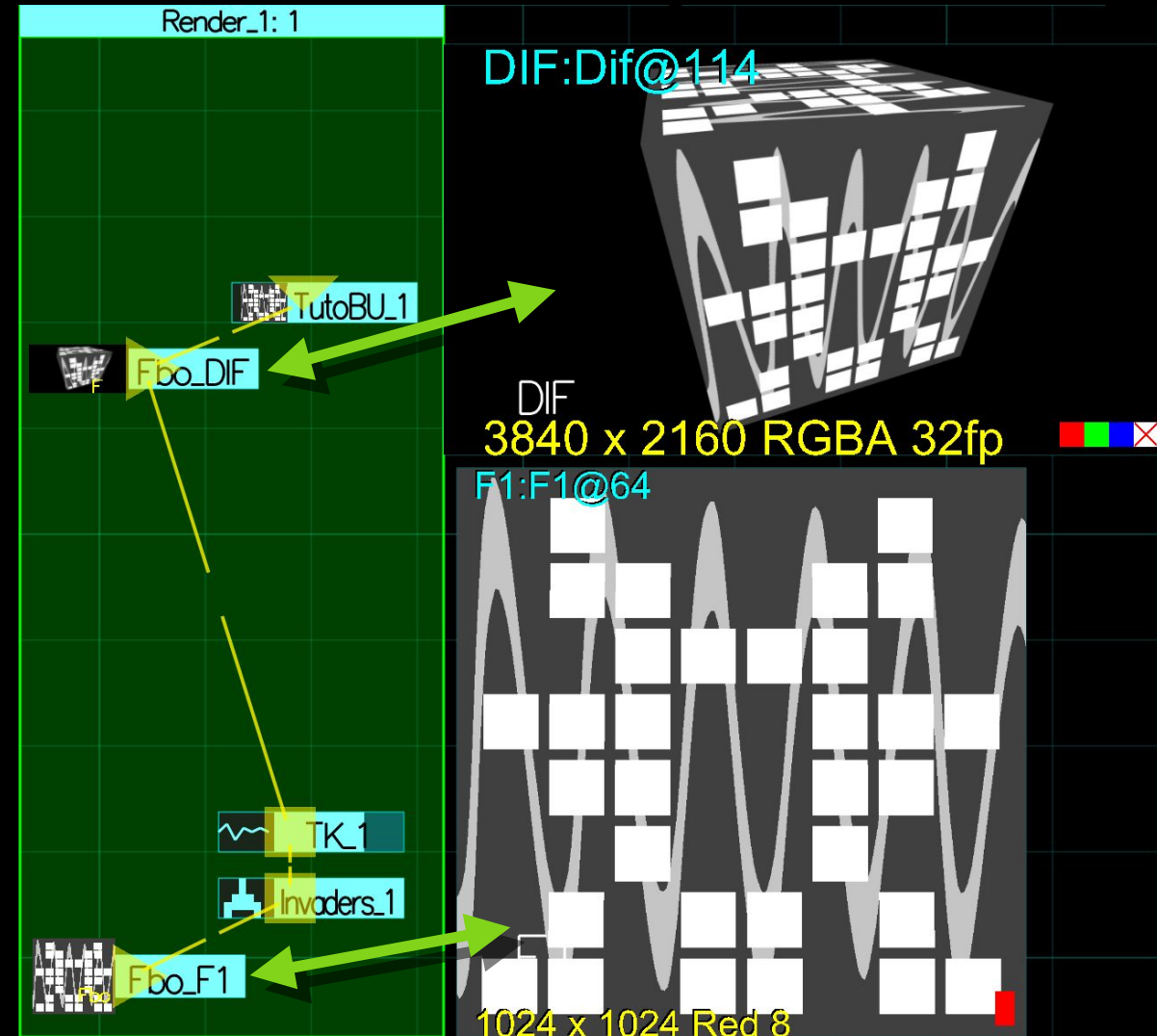
it defines a 4K texture with 4 channels 32fp

it erases itself in black and become the current Fbo

5- Meu TutoBU\_1 is drawn in the current Fbo: DIF

In fact TutoBU\_1 use texture F1 (attached to Meu Fbo\_F1)  
to map on the faces of the cube.

This is an example where a Meu\_Fbo is used to draw to a  
texture (F1) and then this texture is used later by another  
pass.



# MEU Fbo name

- There is two ways to setup a Fbo

## Manually using Flatland

**Not recommended to new users**

- Meu Fbo would eventually get an extended Ui for this

## At creation or renaming

If the **instance name of the MEU correspond to a texture name**  
this **texture is set up as the first color attachment (1)** for the fbo

- **DIF** texture name and **Fbo\_DIF**

**DIF** stands for **DIFfusion**, this is often the Fbo we render at the end and display to the public. It comes from TV control room lingo.

when AAASeed is used live we often have **plenty of graphics processed running**, Some are not even visible to the public, but drawn into Fbos and visualized using BU\_MONITOR or output control monitors. **This way we can prepare, tweak and adjust. At some point we will move a process to be displayed in the DIF.**

Sometimes before displaying the DIF, we add elements (logo, ticker, signature...) to it, or process it with some dedicated MEU (name often Ndc...) to adjust color, reduce the aliasing, add a reference grid...

It is also common practice to have a high definition DIF (4K or 8K) to render, but output at a lower resolution to have a smoother image. This remark apply to any Fbo used for an intermediate pass.

# MEU Fbo Ui Main Tab

- Active → if it is Off: the Fbo is not used
- Focus → Open flatland focused on the c\_fbo object
- Size in pixels

of attachments (all the same)

- Color format
- of color attachments  
in the MEU Ui they are all the same

use flatland if you need more control

- Attachments

active Color (4) and Depth attachment

select which one you see in the monitor below

Local monitor to see the selected attachment

Double Click to open external monitor

Control monitor and texture display

Allow Monitor for attachment 1

Center monitor, flip texture display

Select which Gpu to Cpu transfer happen

when the render chain is done with this fbo, an automatic transfer to the Cpu can be trigger

Select which attachment are saved (see below)

if no selection attachment 1 iss always saved

- Save, in Media AAASnapshots

Depth attachment are not save for now (2025 July)

Save Trig → save One frame

Save → save Each frame

The screenshot shows the MEU Fbo Ui Main Tab interface. At the top, it displays 'Ui4 Fbo\_F1' and 'Main' with a 'Copy' button. Below this, there are tabs for 'Lua', 'Def', 'Both', and 'Loca'. A 'cam\_index' table is visible with indices 1 through 15. The main area shows 'Active' and 'Focus' buttons, followed by '1920 x 1080' and '4 x Int8'. Below this, there are fields for 'Size X' (1920), 'Size Y' (1080), '4 RGB Alpha', and 'Integer 8 bit'. There are also 'Swap' and 'Attachment active' buttons. A table shows attachment 1 as 'Depth'. Below this is a 3D cube visualization with numerical values on its faces. At the bottom, there are checkboxes for 'Monitor', 'Center', 'Flip U', and 'Flip V'. A table shows attachment 1 as 'Depth'. At the very bottom, there are 'Save trig' and 'Save' buttons, and a 'Preset' table with values 2, 4, 6, and 8.

# MEU Fbo Ui Copy Tab



- **Cameras**

sometimes we want all the next MEUs in the render chain to use the same camera. So we can use a camera defined in this MEU.

- **Clear**

control globally if we erase and how we erase

can be done every time or only once by frame rendered (Meu can be used several time in a frame)

## Depth

control erasing of the depth attachment

## Color

control erasing of the color attachments

## Trail

when we don't erase we can have a trail effect. We specify its **length** here.

**Long trail (> 2) require pixel type bigger than 8 bits**

## Alpha

the value we want to set when we erase  
use 0 for transparent background

## Erase Color

a set of predefined colors and a Custom color

**Custom color** components RGB + a grey factor



# MEU Fbo Ui Copy Tab (developer only)

Ui4 Fbo\_F1 Main Copy Fbo <-> Save

S UIUD 0.113ms

Active Focus No

1920 x 1080 4 x Int8

Size X 1920 Size Y 1080 4 RGB Alpha

Size Swap Integer 8 bit

Attachment active

1 2 3 4 Depth

1 2 3 4 Depth

Monitor Center Flip U Flip V

Attachment On Cpu

1 2 3 4 Depth

Attachment to save

1 2 3 4 Depth

Save trig Save

Fbo Nb 1

FlipFlop No Copy Swap

Copy 1 Dst 1 1 1

0.7 1.7 2.7 3.7 4.7 5.7 6.7 7.7

0.6 1.6 2.6 3.6 4.6 5.6 6.6 7.6

0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5

0.4 1.4 2.4 3.4 4.4 5.4 6.4 7.4

0.3 1.3 2.3 3.3 4.3 5.3 6.3 7.3

0.2 1.2 2.2 3.2 4.2 5.2 6.2 7.2

0.1 1.1 2.1 3.1 4.1 5.1 6.1 7.1

0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0

Do Restart

0 -> 1

Preset 2 4 6 8

- Do not touch any of these BUs if you don't know what you do
- This need will be simplified later (2025 August)
- it is used by other MEUs

to render in a succession of textures to keep few frame of history (e.g. do a difference for a camera input with MEU NdcMerge)

deal with fbo « flipFlop » required by some other MEUs (e.g. ReactDif, JumpFlood, Branching)



# AAASeed

## An introduction

### Part 17: Meu Monitor

- Display Fbo attachment
- Copy Fbo attachment
- MEU Ui
- As a “texture plug”



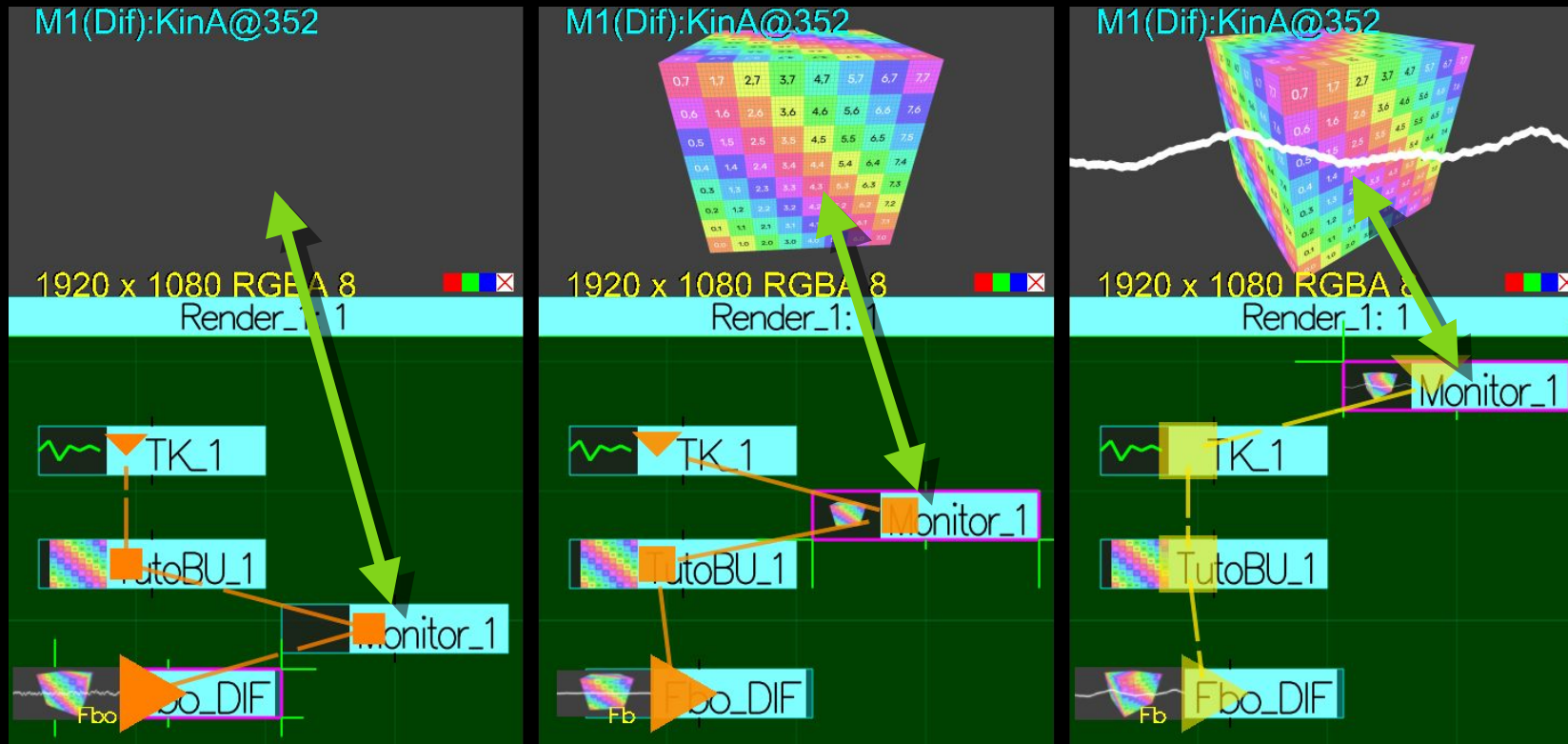


# • MEU Monitor Display Fbo attachment

- Display state of current Fbo in the render chain

See what is going on in the render chain

Provide control monitor at different stage in the control chain



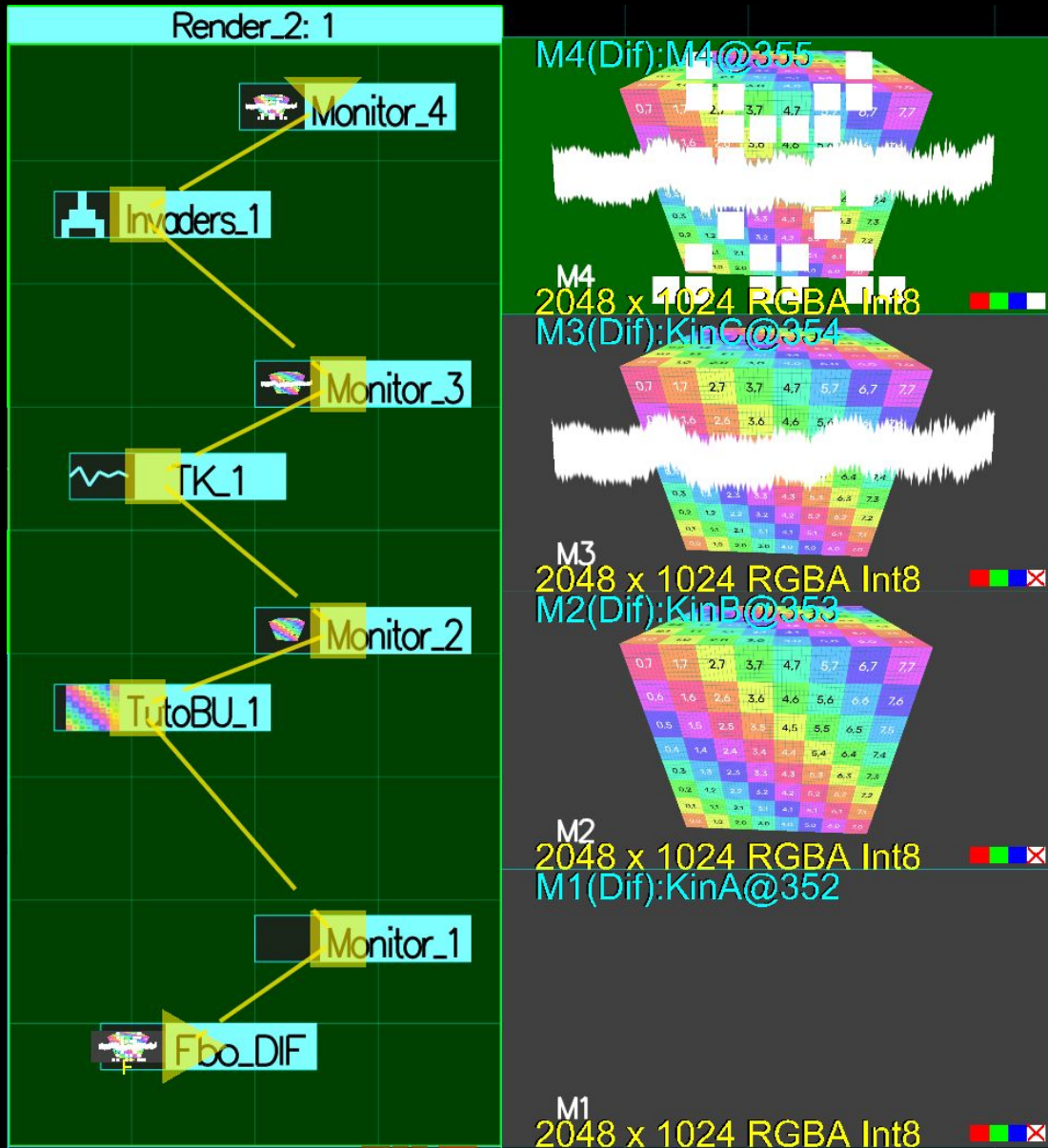
- Copy texture attachment in its current state to a texture associated with the MEU Monitor

# • MEU Monitor Copy Fbo attachment

- Here 4 MEUs Monitor let us see how MEUs in the render chain draw one by one in the same fbo.
- In fact each MEU Monitor also save a copy of the attachment (texture) in its own texture bind

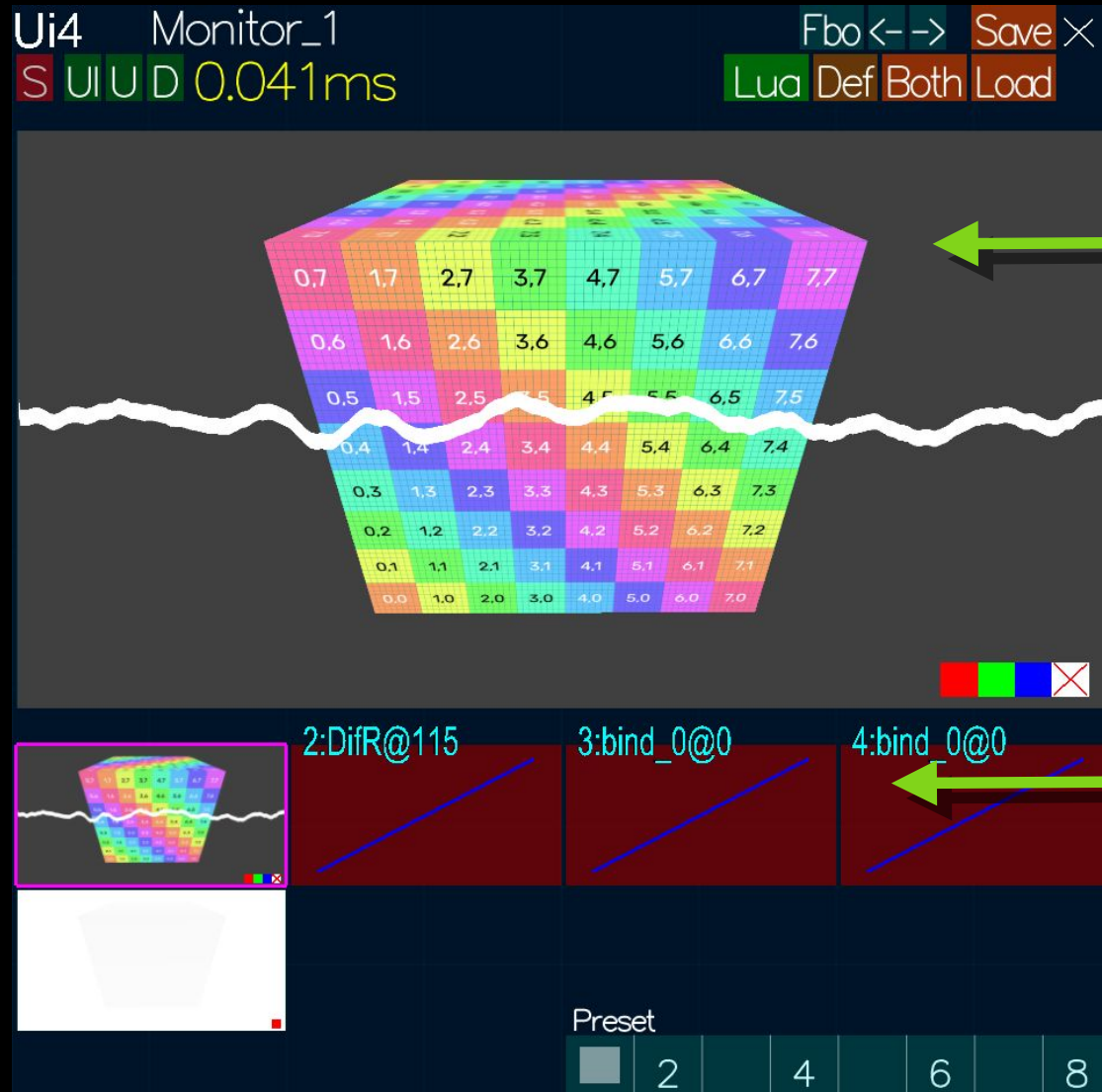
M1, M2, M3 and M4

352, 353, 354 and 355 in term of flat bind.





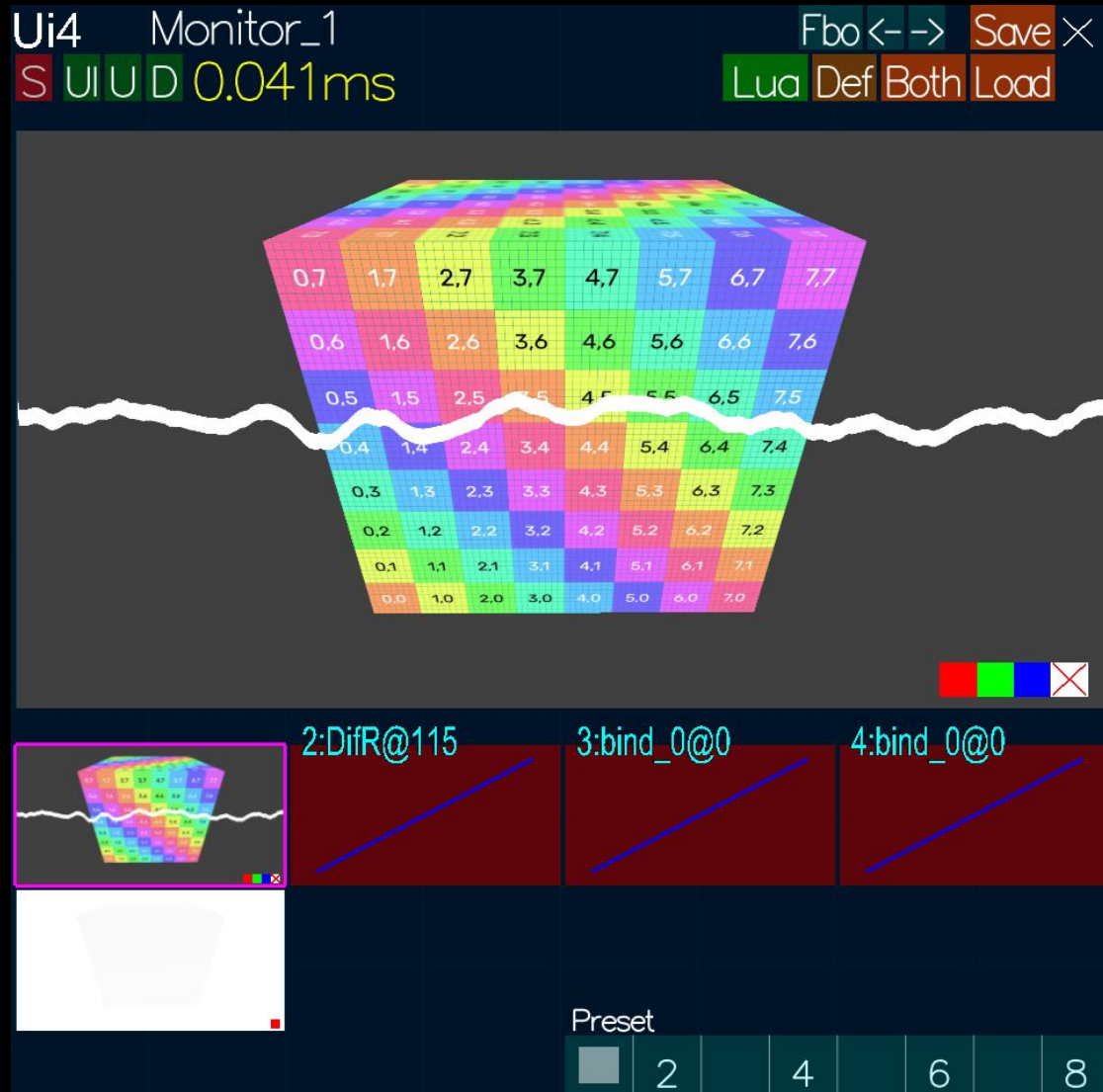
# MEU Monitor Ui



- Show Selected attachment  
Double Click open associated monitor.

- Select which attachment is copied and displayed

# MEU Monitor as a “texture plug”



- It really does texture copy  
yes it have a cost  
Copy time  
Done at low level on Gpu  
fast so low cost  
Synchronization
- Destination is texture named M1 to M32  
Monitor should be named Monitor\_1 to Monitor\_32
- Destination texture can then be used  
so the Fbo attachment copy can be used  
by other meu later in the render chain.  
The Monitor's MU position in the render  
chain can be changed on the fly, picking  
the texture we want without altering the  
rest of the render chain.  
So we can “plug” any stage of the render  
chain as input for others Meu.  
This is “an input texture plug”

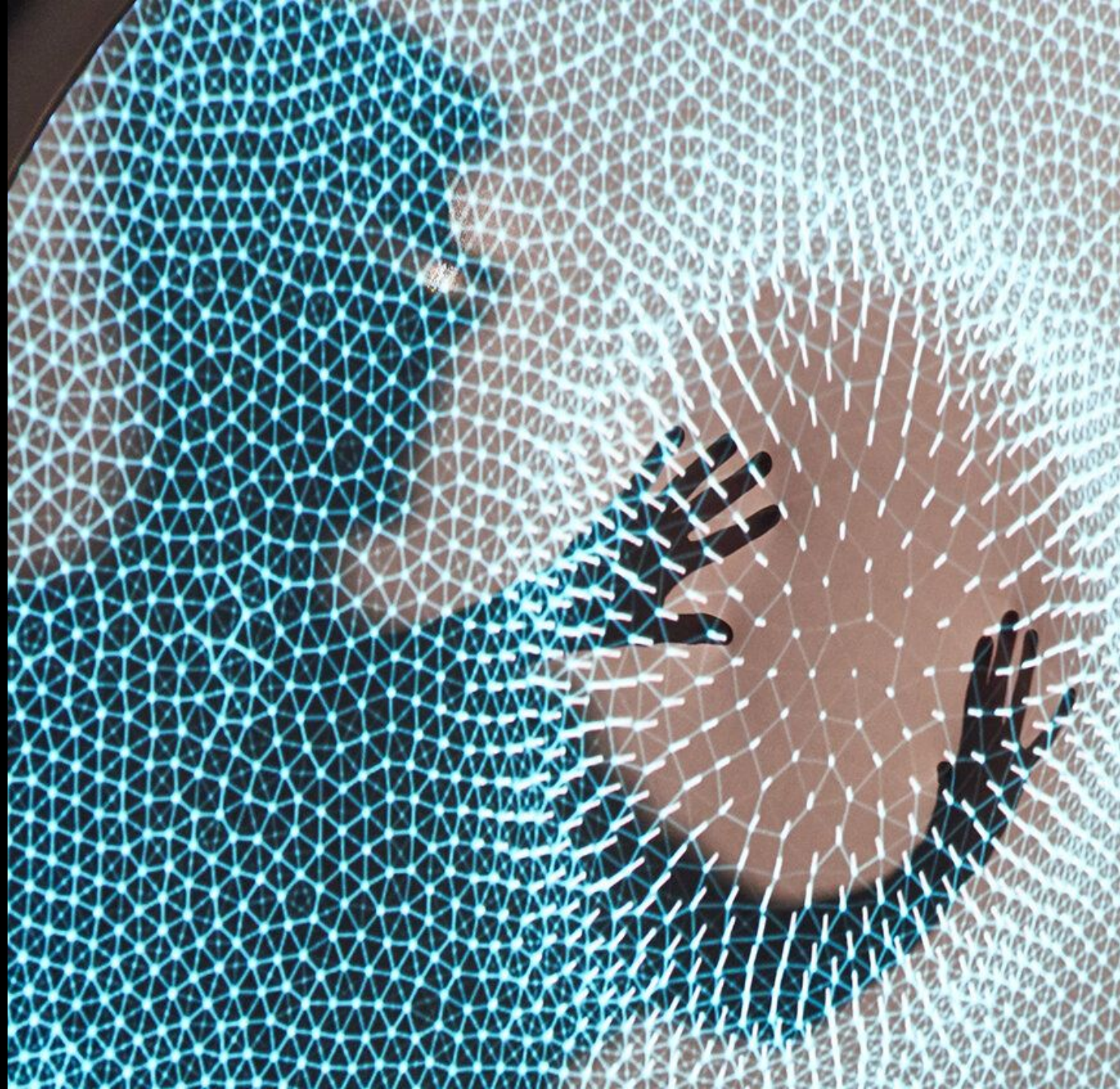


# AAASeed

## An introduction

### Part 18: Create MEU

- Base MEUs
- MEU Window
  - **Info and selection**
- MEU Import





# Base MEUs

- The ones we already describe in this introduction
  - Dir → Encapsulate MEUs
  - Fbo Frame Buffer Object → Canvas where we draw
  - Monitor → Display and copy the current state of the current Fbo
- The ones we describe in rest of this introduction
  - Video → Decompress video in a texture
  - PIP Picture In Picture → Display a texture
  - Out → Output a Texture for screens, multi-screen, Led wall ...
  - TRAX → Plug BU in and out
- The ones you need
  - Depends on your field, what you want to do
- The ones to learn
  - MEU Tuto...
- The ones you make
  - AAASeed let you customize and/or add your own MEUs
  - Another document to come



# MEU Window

- **Navigate MEU**
  - Mouse Wheel anywhere
  - Scroll list
  - Mouse over item
  - Display Type Info
- **Click on item**
  - Open StarMenu
    - 
    - New
      - Create a new instance of this type
      - A dialog let you choose the instance name.
      - The new instance will appear under the Unused button.
      - 
    - Lua
      - Open the lua script for this type
    - Dir
      - Open with the file explorer the directory where the prototype is.
  - Double Click on item
    - New like StarMenu

Meu

Proto

Self

All

Proto

No Tag

All

Tag

2d		
3D		
Art		
Camera		
Core		
CoreGraphic		
Deprecated		
Device		
Draw		
Experimental		
Generator		
Geometry		
ImageProcessing		
Input		
Interoperability		
Output		
Point		
Procedural		
Proprietary		
RenderPass		
Sound		
Surface		
Text		
Texture		
Tutorial		
Unfinished		
Utility		
VJ		

Fbo(Frame Buffer Object) by Maa

MEU nb 134/142

This is a canvas:  
Fbo define a group of textures to draw to (attachments)

TAGS: 2d Core CoreGraphic Draw Texture VJ

2054	AAASeedShoo	AAASlide	AAATree
AAAUtils	AIChatGptLouis	App	BlobDetect
Blur	Boid	Bullet	Cam
CaptureRect	CartoMaton	CartoMatonController	Clear
Cloud	CIPool	ColorCurve	DepthPick
Derviche	DigitalProjection	Dir	Displace
DisplacePart	DistField	ExShaderGrid	ExShaderInstance
FaceTrak	FaceUV	Fbo	Fbx
FbxMatte	FieldGene	Flex	FlexVideo
FP	GeoUV	Grab	GridSel
Hexa	HexCraze	ImgAnal	ImgAnalMulti
ImgSend	ImgTintMulti	Invaders	Kinect
Kinect1	KinFlipper	KinMove	KinMoveAuto
Lidar	LightPassV1	Lights	Marseille

Use

Tuto

Ui4

# MEU Window

Self	All	Proto
Togs		
No Tag	All	Tag
2d		
3D		
Art		
Camera		
Core		
CoreGraphic		
Deprecated		
Device		
Draw		
Experimental		
Generator		
Geometry		
ImageProcessing		
Input		
Interoperability		
Output		
Point		
Procedural		
Proprietary		
RenderPass		
Sound		
Surface		
Text		
Texture		
Tutorial		
Unfinished		
Utility		
VJ		

## Info zone

it displays information defined by MEU type

Type name (Type long Name)

Name by author

Date and version

Documentation

TAGS for this type (used for selection)

- number type shown / number type existing

## Select by Tag

selection is all Tags active (On/Green)

Button at right of Tag button

On/Green → tag wanted

Off/Red → tag excluded

Here we want Types with  
(Tag Camera) and (not Tag Deprecated) and Tag Device

button Red/Green at top set all button under

## Select by Name

activate the button Use

## Proto type selector

**Self** → show only types for which an instance is its own prototype. We talk about **self or local prototype**.

**Proto** → Show types which have a prototype inside a MEU\_PROTO folder. We talk about **isolated prototype**.

**All** → show all: don't select on prototype local or isolated

## Tag selector

**No Tag** → select only type for which no tag are defined

**Tag** → select only type for which tag are defined

**All** → show all: don't select on presence of tag or not

Fbo(Frame Buffer Object) by Maa

This is a canvas:

Fbo define a group of textures to draw to (attachments)

TAGS: 2d Core CoreGraphic Draw Texture VJ

MEU nb 134/142

Camera	
Core	
CoreGraphic	
Deprecated	
Device	

No Tag	All	Tag
2d		

Use Tuto

Proto		
Self	All	Proto
Togs		
No Tag	All	Tag

# MEU Import

- to create MEU we know already
  - 1/ Use MEU Window (previous 2 slides)
  - 2/ instantiate an existing one
    - Use StarMenu
    - Drag a MU and release with Control Key pressed

- We can also import a MEU for another APP

Use Main Menu

a File dialog is opened

Navigate to any APP folder

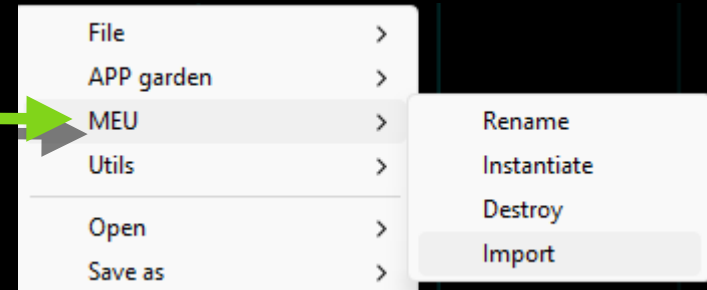
Open folder AAA\_MEU

eventually if it a MEU\_DIR folder you will find again an AAA\_MEU folder inside...

Select a folder

Validate

a copy of the directory is added to your current App and the instance Appear under the Unused Button





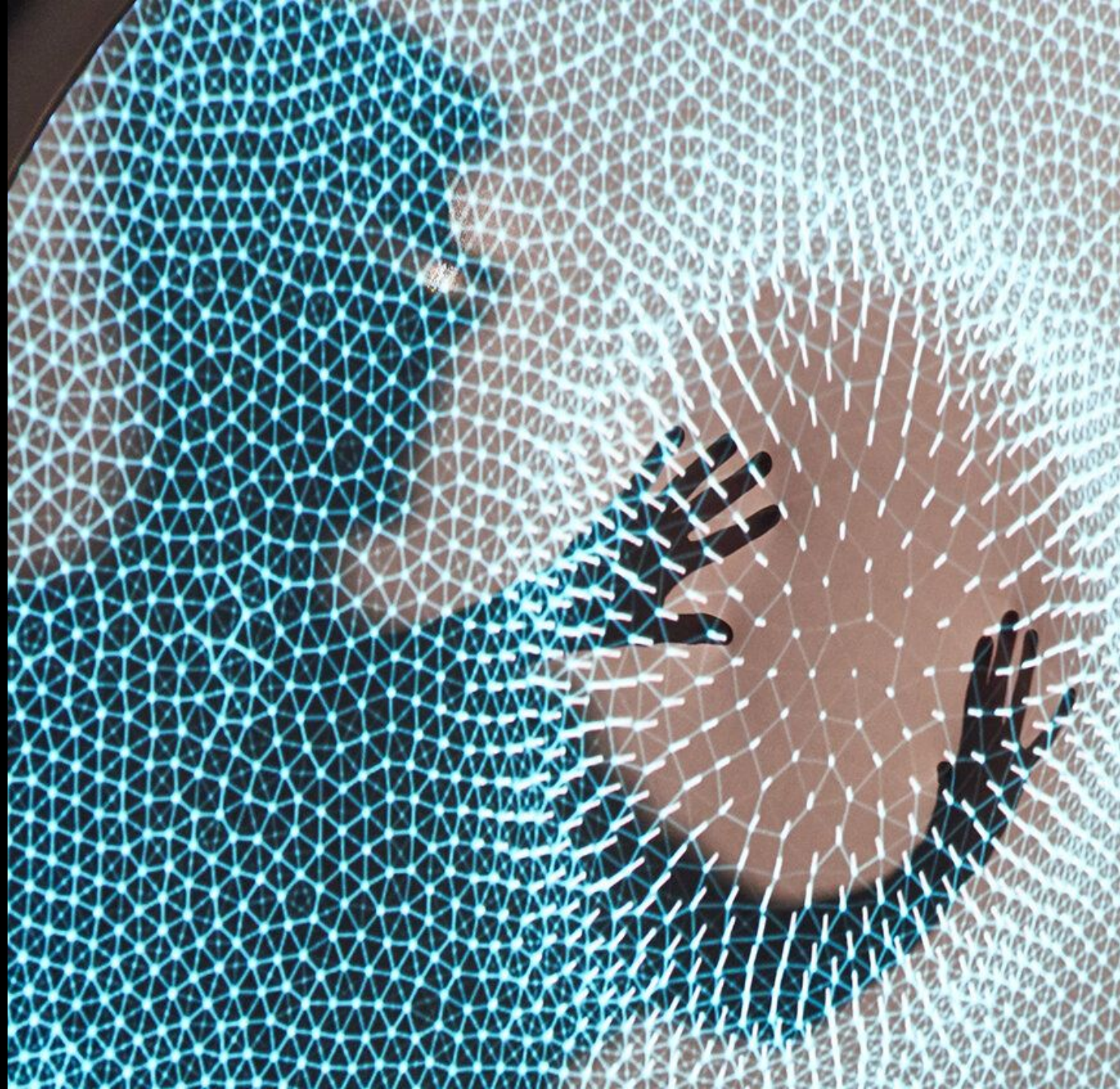
# AAASeed

## An introduction

### Part 19

### Reading video

- **Meu Video: Send an Image stream to a texture**
  - **Read video File**
  - **Capture**
- **Meu Video**
- **Gain Bias**



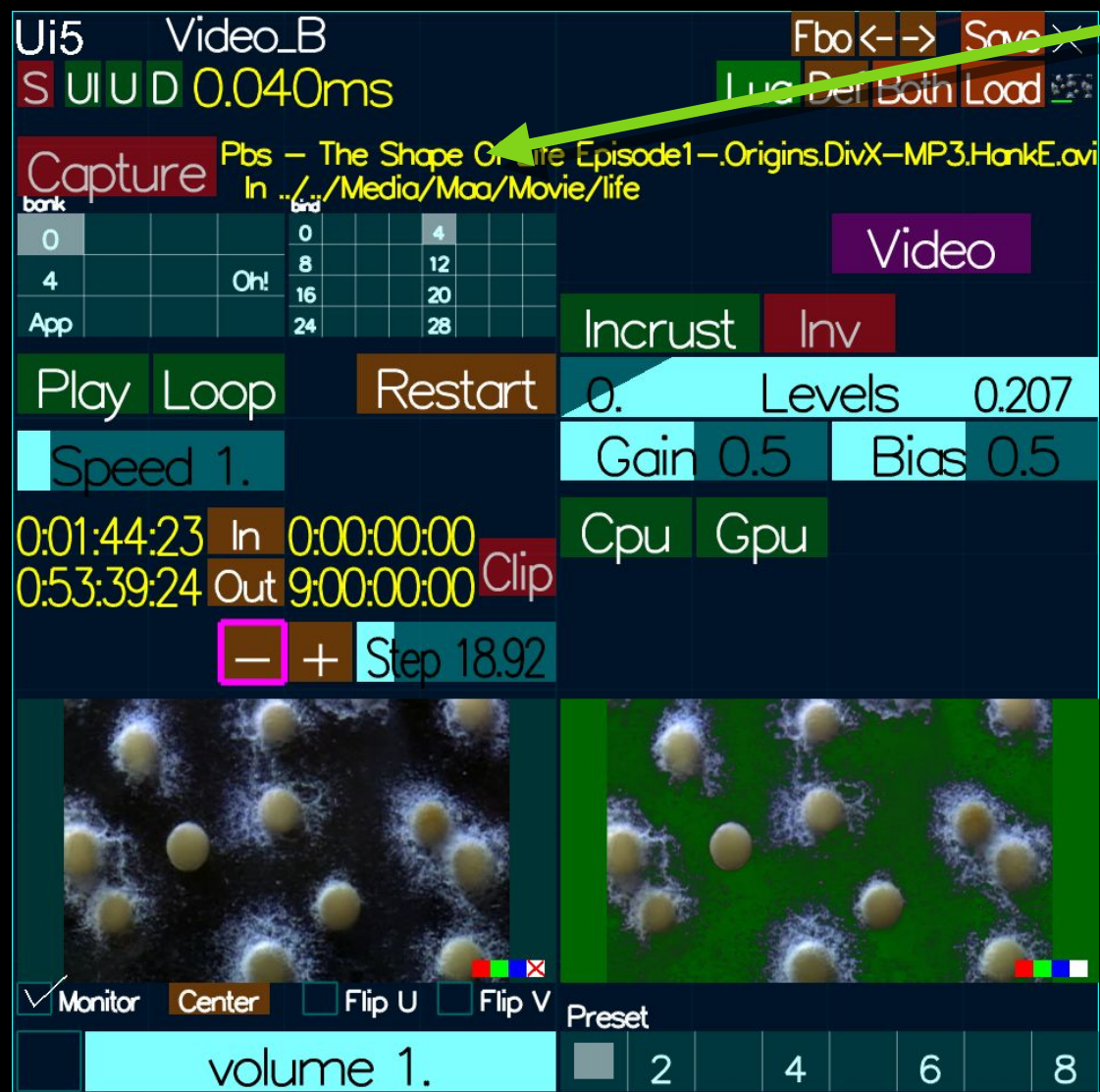


# Meu Video: Send an Image stream to a texture



- MEU\_Video deals with an image stream:
  - It decompresses a video file
    - It uses different method for that and codecs need to be installed for that (See part 2 Install).
    - Video can be started, stopped, looped, skipped, read at different speed, clipped...
  - It captures a live input
    - Configuration using flatland
      - Not the object of this document
    - Cameras
      - webcam
      - industrial
      - Kinect V2/Azure (as regular camera), PS3....
  - Capture devices
    - Capture cards
    - Usb capture cards devices
  - Display Grabber
    - Can grab any part of windows screens: any software can be an image input
  - done on the CPU side
- MEU\_Video does incrustation
  - Luma (Luminence) incrustation for now (2025 September)
  - Color incrustation done by MEU PIP on the Gpu side
  - done on the CPU side too
- MEU\_Video transfer the image to a texture
  - a CPU to GPU transfer
- MEU\_Video have an associated external monitor (A to L)
  - Can be setup in the Ui

# Meu Video



## Video Name and folder

Red when there is a problem: look at terminal window

- Bank / Bind like texture

12 banks of 32 binds

2 selectors

Click double

Pick a video for the video bind

bank	bind
0	0
4	8
App	16
	24
	28

- Play / Control

In / Out / Clip → play a subpart

- / + → move in time by Step increment

- Monitor (Ui as MEU\_Fbo)

- Volume

On/Off button and Slider

There is a master Volume Too

- Capture → switch to capture mode

more in a separate document

- Video → Focus on c\_tex\_video in Flatland

- Incrustation

Luminance Incrustation

Inverse → inverse the curve

Levels → min max levels of incrustation

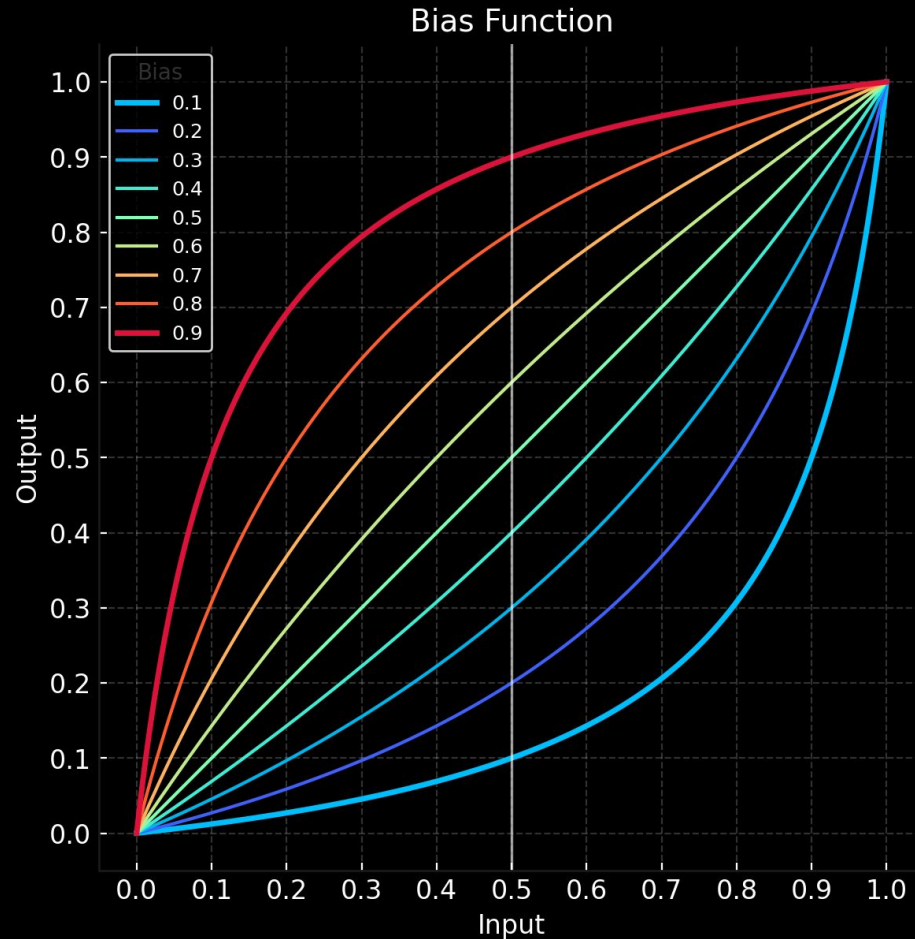
Gain / Bias alter the shape of the transfer curve (see next slide)

- Cpu and Gpu

Where we keep it



# Gain Bias an electronic heritage

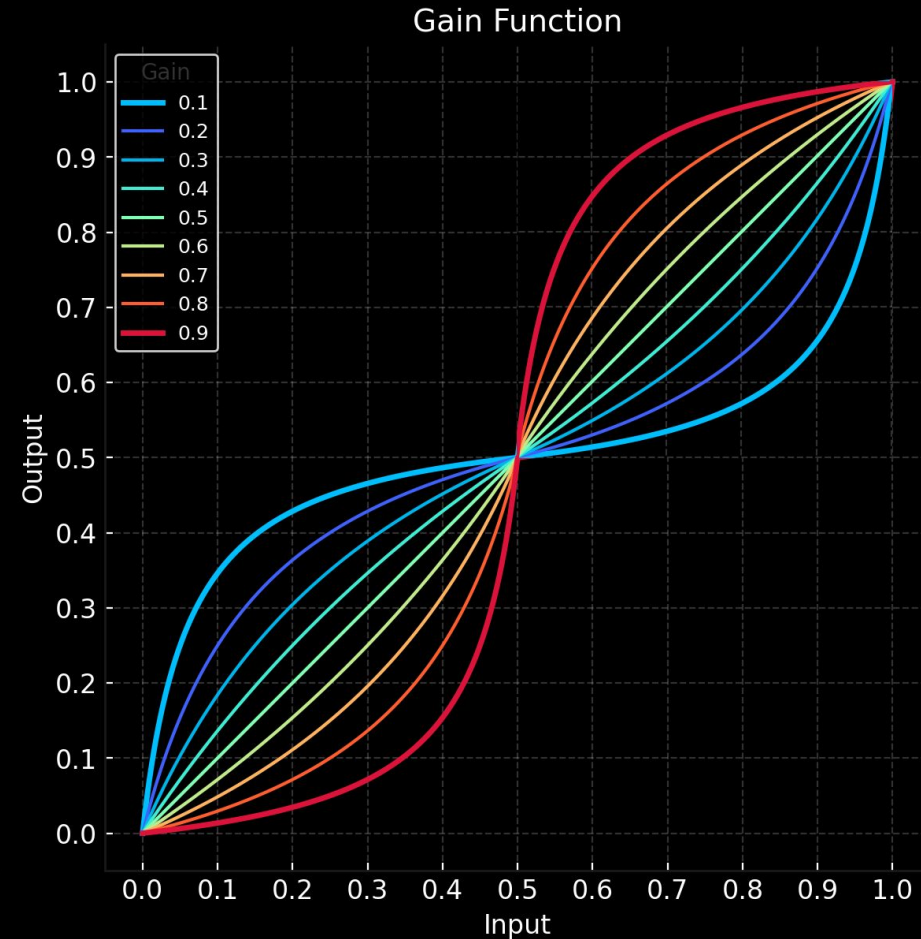


Input at 0.5 : Bias value

Low Bias → Push values down

0.5 → Straight curve

High Bias → Push values up



Double Bias curve (symmetry **in the middle**)

**Low Gain** → Push **value to middle**

0.5 → Straight curve

**High Bias** → Push **values to extreme**



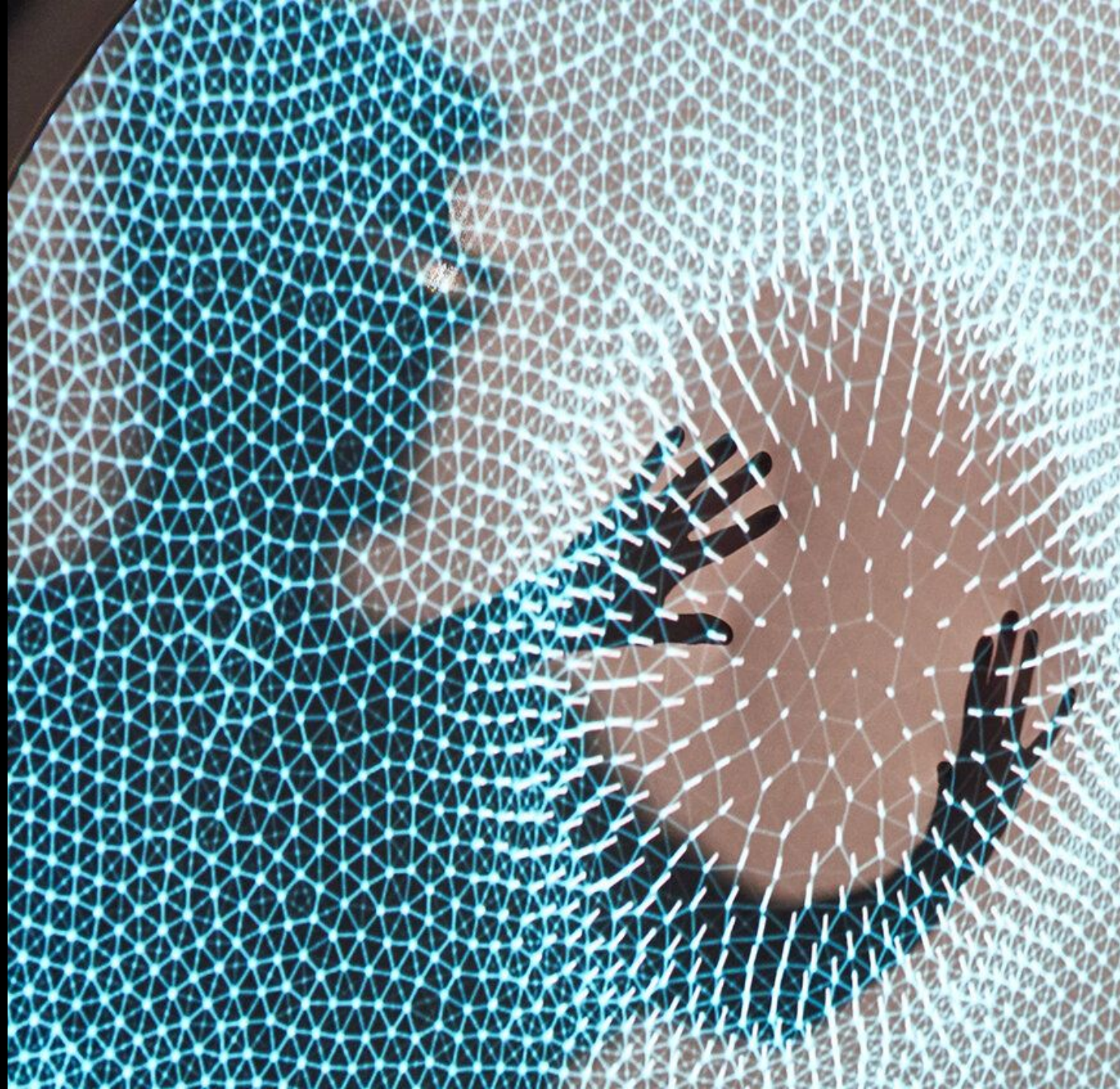
# AAASeed

## An introduction

### Part 20

### Drawing texture

- Meu PIP: Picture In Picture
- Main tab left
- Main tab right
- Ratio and Size
- TRS
  - **Translate Rotate Scale**
- Map tab
- Blending





# Meu PIP: Picture In Picture

- **Picture In Picture**  
from the **video world**  
Drawing a video in a rectangle
- **MEU\_PIP draws a texture in a rectangle**
- **MEU\_PIP does much more**
  - Crop the texture used**
  - Handles smooth transition of texture**
  - Performs Incrustation on the Gpu**
    - Luma (Grey)
    - Color
  - Composes with a the texture mask**  
using its **alpha or a Remap** (luma incrustation)
  - Sizes the rectangle**
  - Transforms the rectangle in 3d**
  - Recolors the output**
  - Clamp the color**
  - Add a fuzzy edge**
  - Composites with the Fbo**  
Does even more, but not the coffee or tea
- **Part of its Ui are generic and used in other Meu type, and will be detailed here**



# Meu PIP Main tab left

- **Tex** selects next Texture to be displayed
- **Inverse** → Inverse the RGB colors (**negative**)
- **Incrustation**

**Incrust** → toggle Incrustation

If active the **alpha produced**  
replace the texture alpha

**2 modes** to produce the alpha values

**Luma** (Luminance)

use the grey from the texture

**Color**

use the distance to the color

**Luma SLIDER TWO** define the **curve** used to transform  
grey or color distance to alpha

**Inv** Button **flip** the **Luma Min/Max** values

- **Swap** → swap texture and mask

- **Mask**

**Mask** → toggle mask use

Alpha used is the **multiplication** of texture alpha  
and mask alpha

**Remap** acts like a **luminance(Luma) incrustation** for the  
mask

**Mul** means **Multiply** it activates the **multiplication** of  
texture **rgb** by **mask rgb**

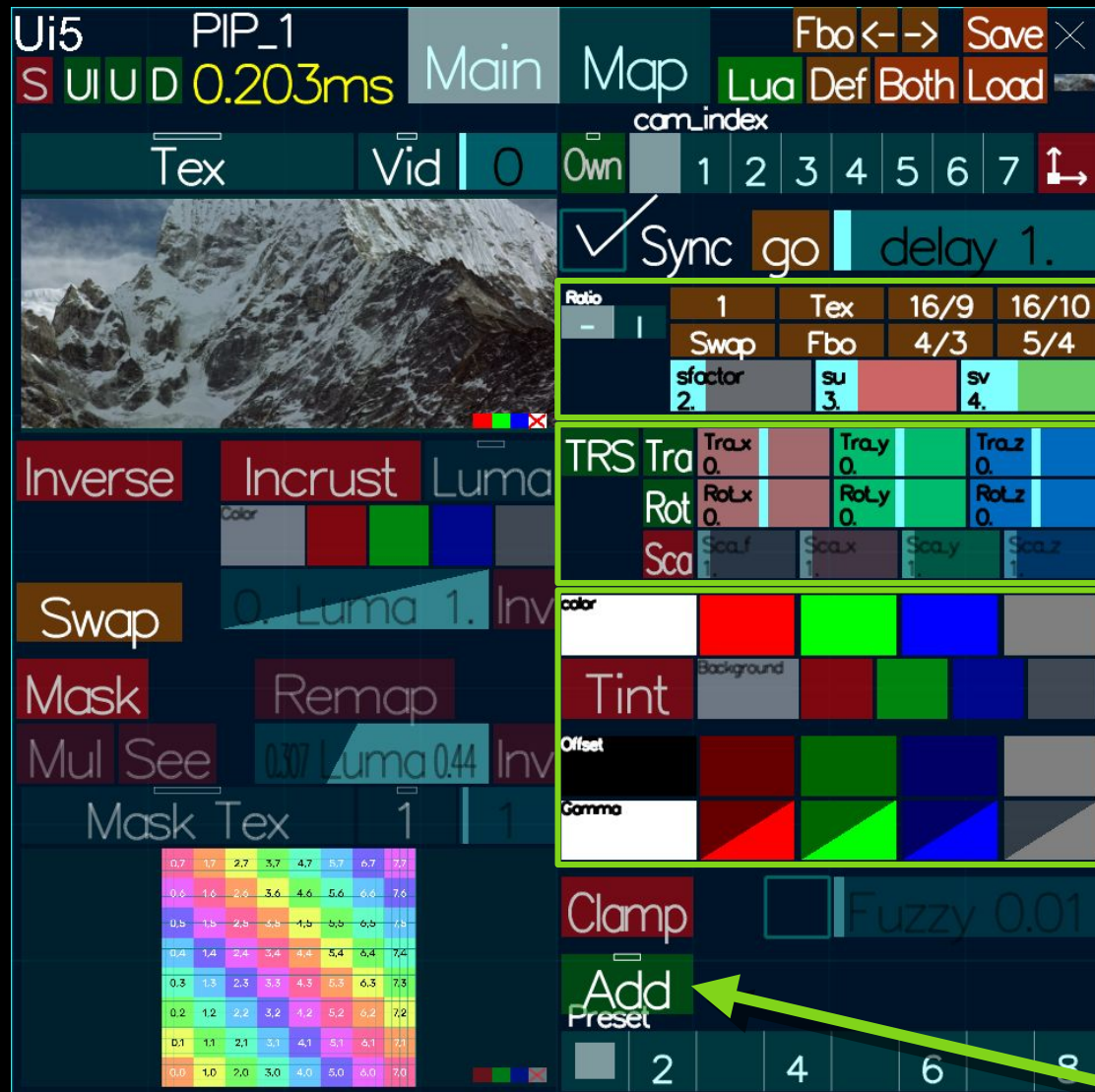
**See** button shows the mask, ignoring the texture

**Mask Tex** selects texture used for mask





# Meu PIP Main tab right



- **Cameras**

The Usual, See Part Cameras & Axes

- **Sync**

if **active** the texture **transition is automatic**

**Go** triggers manually the transition

**Delay** control the duration of the transition

- **Size of the rectangle** (see next slides)

- **TRS** (see next slides) move the rectangle

- **Color manipulation**

**Color** multiplies the texture color

**Tint** transforms the texture **grey** into an interpolation between background color and color

**Offset** is added to output color

**Gamma** performs a gamma transformation to the output color

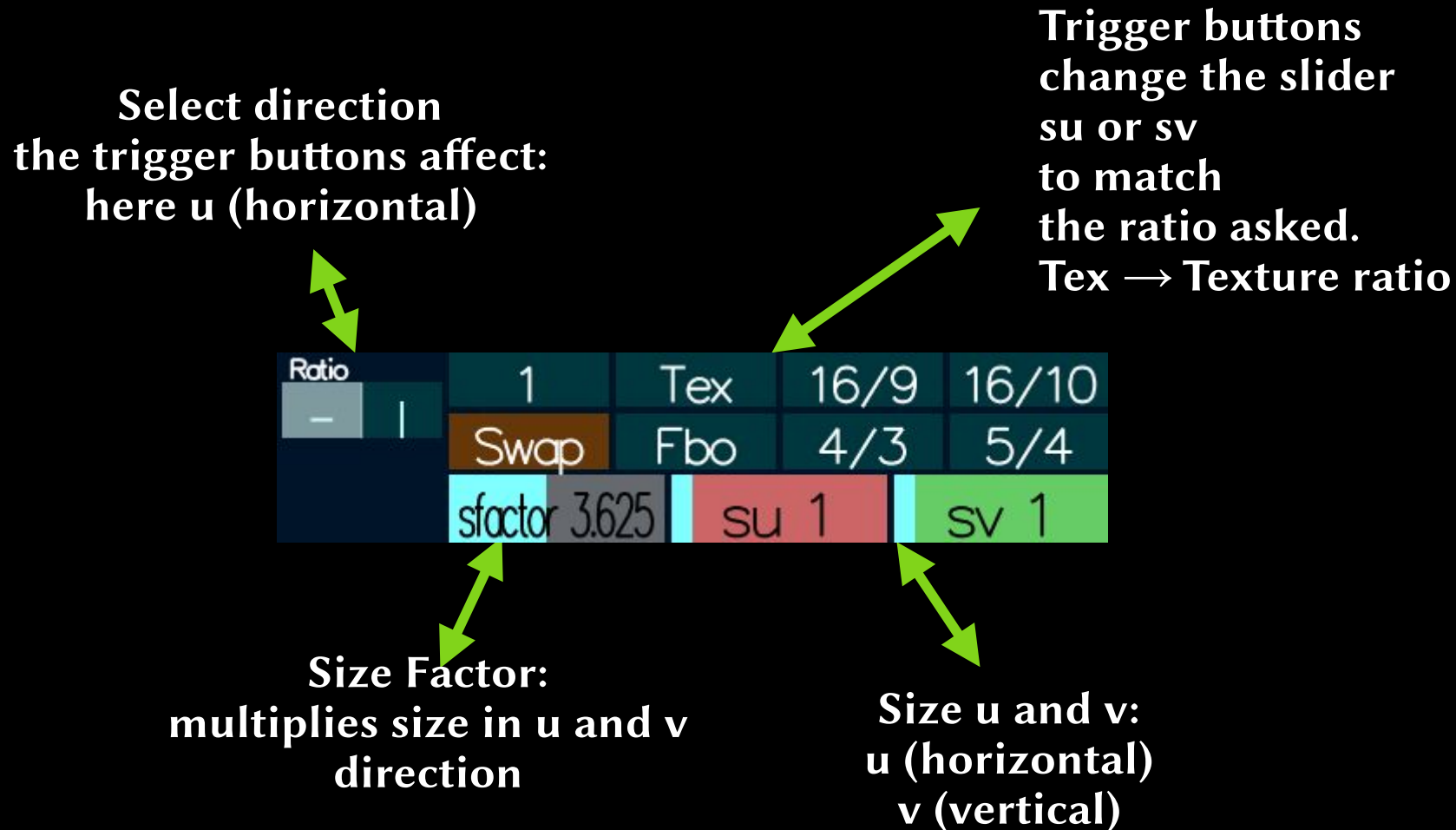
- **Clamp** limit the color output to the **[0,1]** interval

- **Fuzzy** adds a transparent border at all edges

- **Blending** (see next slides)

# Ratio and size (used in other MEU type)

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





# TRS ↔ Translate Rotate Scale (used in other MEU type)

- This controls what is named in 3D: a **transformation**

- **Changing** where an **object** is drawn

- Not moving the camera
- Developer note

Alter the model (model to world) matrix

- **Combine 3 Types of transformations**

- **Tra** → **Translate**

**position of the center**

- **Rot** → **Rotate**

**rotate around the center**

Rotation unit in AASeed is trun/revolution

0.25 → a quarter turn

0.5 → a half turn

1 → a full turn

- **Sca** → **Scale**

**Scale around the center**

**Sca\_f (Scale factor) multiplies individual axe values**

- Better **keep button TRS Off** if transformation is **not used**

**General button**  
allow global TRS use

**Individual buttons**  
allow each tranformation



# Meu PIP Map tab

- **nb\_u, nb\_v**  
the textured **rectangle** drawn by Meu PIP is in fact a **grid**. This controls the **numbers of points** (and so subdivisions) of the grid in both directions
- **U Min/Max, V Min/Max**  
controls which part of the texture is used
- **Hexa**  
**Fun mode to map with hexagonal symmetry**: please experiment  
**du, dv**  
**offsets**
- **UV Bdd Src**  
Allow to use **UV mapping generated** from a bdd, usually a **c\_bdd\_grid\_adjustable** used to edit a deformation in conjunction with an source image. **See Meu Out in next Part.**  
**Uncheck** if you don't know what you do



# Blending (used in other Meu type)

- This small and perhaps too discrete BU controls **Blending**



- Blending**

is how the pixel produced by the MEU,  
called a **fragment** at this stage,  
is applied to the Fbo.

In fact it controls what is called the **blending equation**

- it is applied channel by channel : R,G,B and Alpha

**Min**: use the **Minimum** between what you draw and what is in the Fbo

**Max**: use the **Maximum** between what you draw and what is in the Fbo

- **Add**: use an **Addition** between what you draw and what is in the Fbo  
using Alpha to do transparency  
$$\text{destination} = \text{fragment} * \alpha + \text{fbo} * (1 - \alpha)$$

**This is the default mode and what you should use in general**

**the blending BU are drawn with a discrete flashing when value is not Add**

**Sub**: use the **Subtraction** between what you draw and what is in the Fbo

**RSub**: use the **Reverse Subtraction** between what you draw and what is in the Fbo

- Other modes correspond to photoshop compositing modes** : please experiment
- More on blending modes at interactive web Blending application

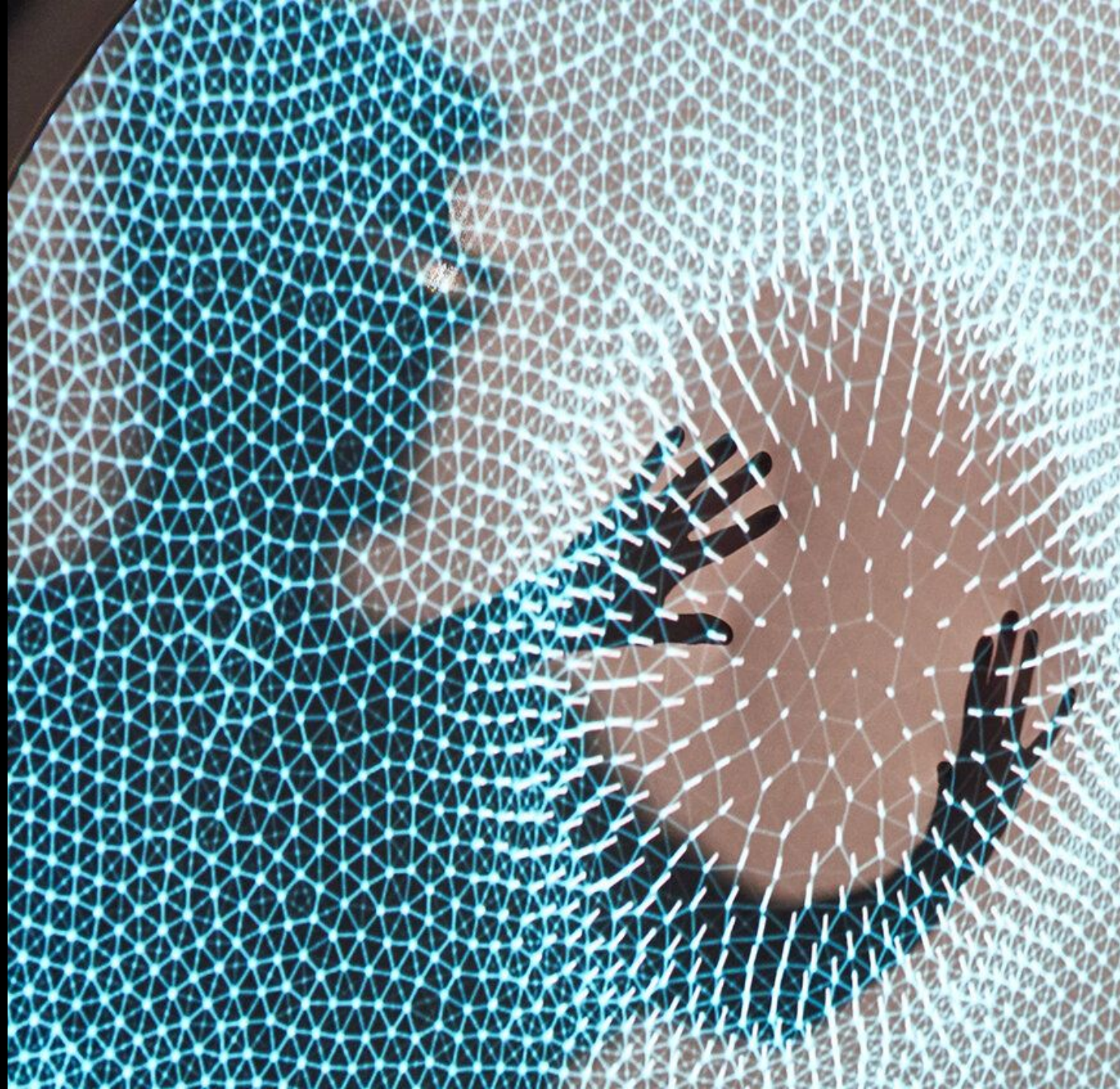


# AAASeed

## An introduction

### Part 21: Outputting image

- MEU Out
- Main tab
- Edit Deformation Grid
- Fuzzy edges
- Map tab
- Rendering BUs





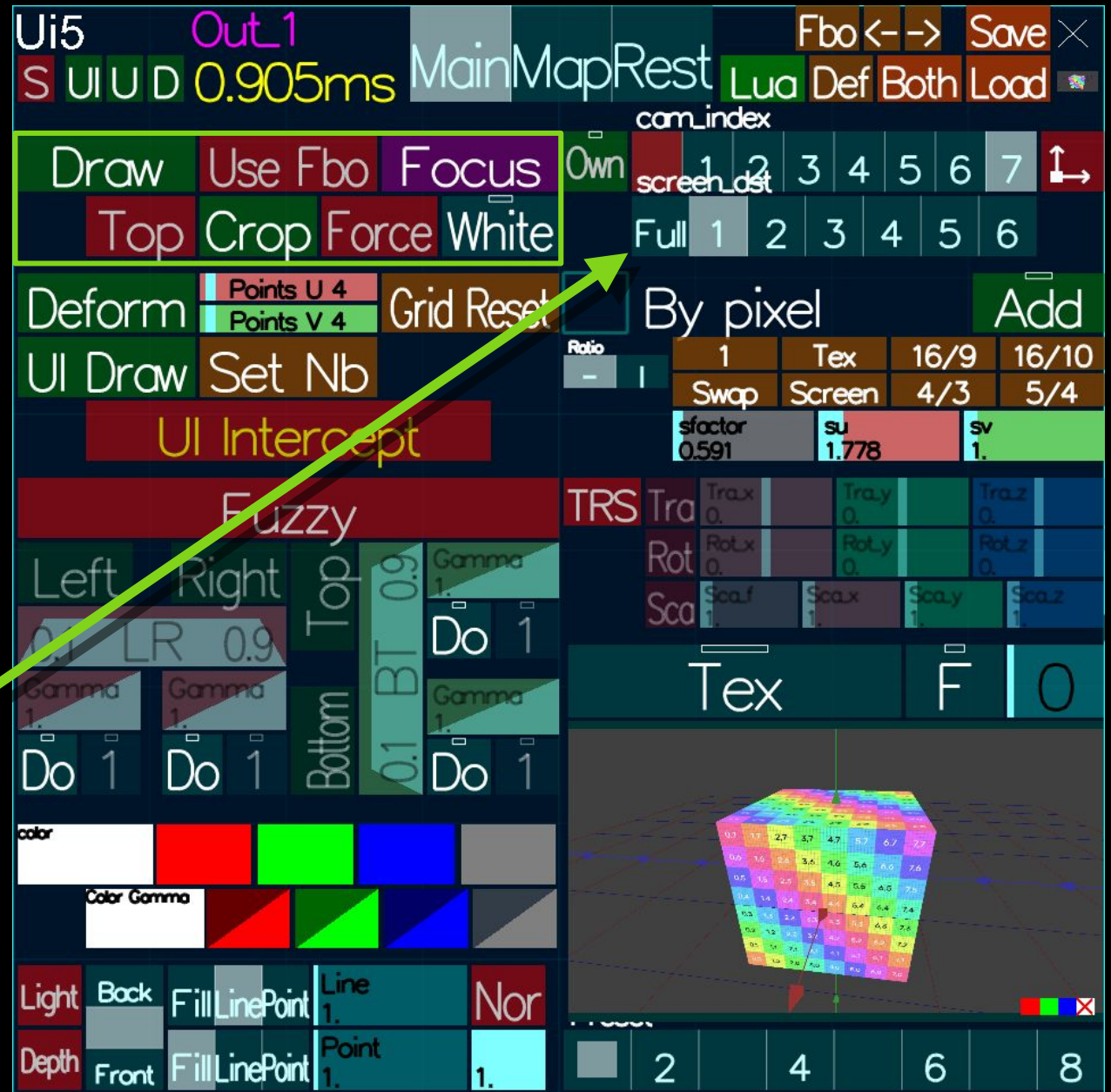
# MEU Out

- **Handles Final output**  
to screens, projectors or LED walls
- Its main function is to **set up an Output quickly**
- **MEU\_Out** like MEU\_Pip draws a texture in a rectangle
- **MEU\_Out**  
 Crop the texture used (tab Map in next Slides)  
 Draw the texture to a selected screen (Windows screen)  
 Size the rectangle  
 Deform the rectangle  
 with a **real time editor** made for difficult conditions  
 Output pixel exact image  
 to deal with **Led walls**  
 Add fuzzy edges (feathered, blended edges)  
 editable for each edge  
 Can also draw to an Fbo  
 nice to have **deformable editable grid** there  
 Composites the outputs  
 Can be used as a texture UV Editor too
- Part of its Ui are generic and used in other Meu type, and will be detailed here



# MEU Out Main tab

- **Draw** should be **On**  
Or **Off** if the deformation is used somewhere else but should not be displayed (document on this will come)
- **Use Fbo** button  
Off → draws to screen  
On → draws to current Fbo
- **Top** show the **grid drawn** on top of the texture drawing  
this is the drawn grid **not the deformation grid**
- **Crop** use the **crop defined in the Map tab** (see next slide)
- **Force** forces the **display** according to the **selector to its right**  
quick way to switch the output  
you can see better what you do  
projector can be adjusted ...
- The selector under the camera (**screen\_dst**) controls to which screen the meu draws  
Full means all the window  
numbers correspond to windows Display Settings  
Here we select screen 2



# MEU Out Main tab Deformation

- **Deform** activates the deformation grid and process

When **Deform** is **On**, the deformation grid take over so any modification in the ratio and size part of the Meu have no effect.

To change the deformation grid size

set **Deform** to Off

adjust with size and ratio BUs

push **Grid Reset** and confirm

- **Set Nb** button change the subdivision of the deformation grid

using the sliders values (**Points U/V**)

- **Grid Reset** resets the deformation grid after a confirmation

- **Focus** open **Flatland** focused on the **c\_bdd\_grid\_adjustable** used

- **UI draw** show the grid editor **Ui**

In **Blue** when edit is inactive

in **Red and Green** when edit is active

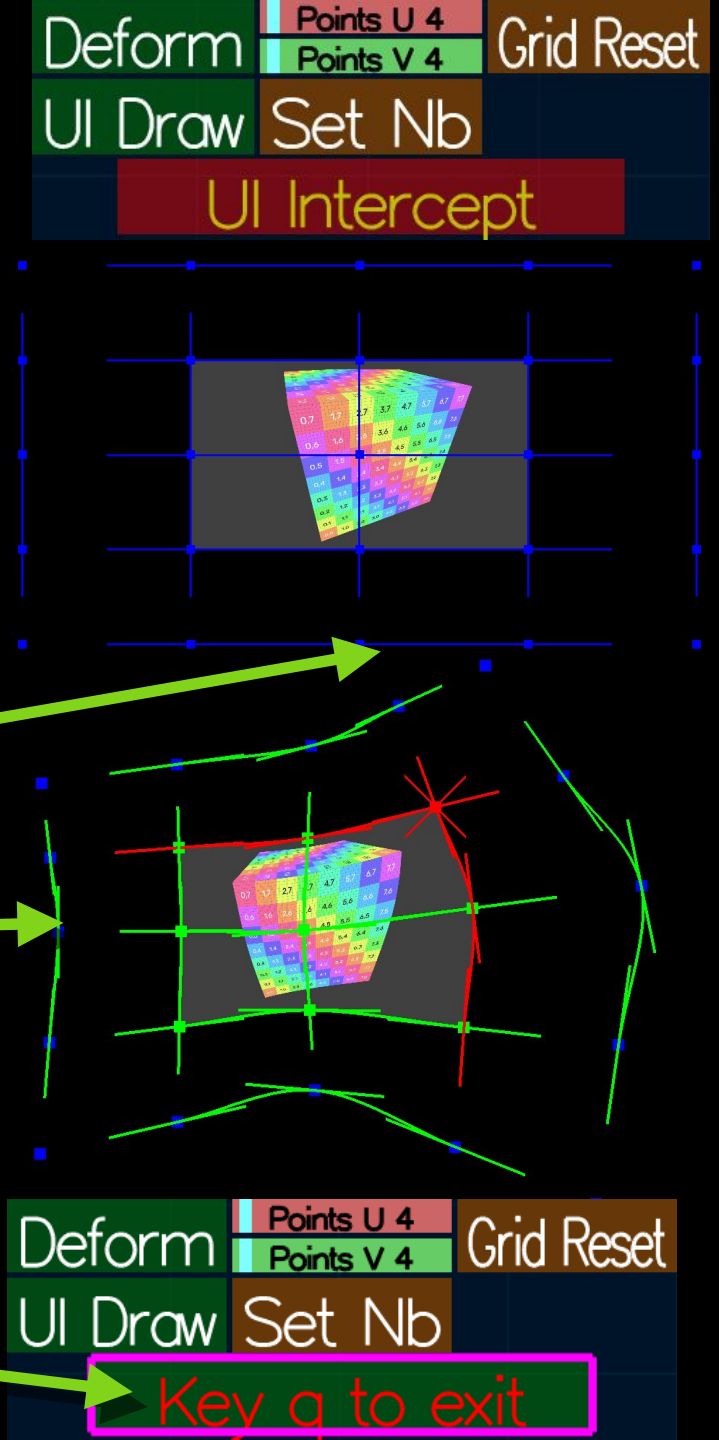
Adjust **Line** and **Point** size to your context (sliders at bottom)

- **UI intercept** switches the grid in **Edit mode**

**GaBuZoMeu** Interface is not usable but **Flatland** is  
use **F2** to hide/show **GaBuZoMeu**

Key **q** → exit this mode

and then enter edit mode back





# MEU Out Edit mode

- Arrow Keys → change the current point  
the one at the intersection of the two red lines
- Click the mouse and Drag → move the current point

Some points (in Blue) can not be edited, most of the time because they are computed automatically. Here the external points.

Ctrl and Ctrl Alt → move more precise

Shift Ctrl and Shift Ctrl Alt → move amplified

Ctrl z/y → Undo/Redo (just last move) when no flatland.

Ctrl r → Reset the deformation grid

- Keys

q → (q)uit the edit mode and then enter edit mode back

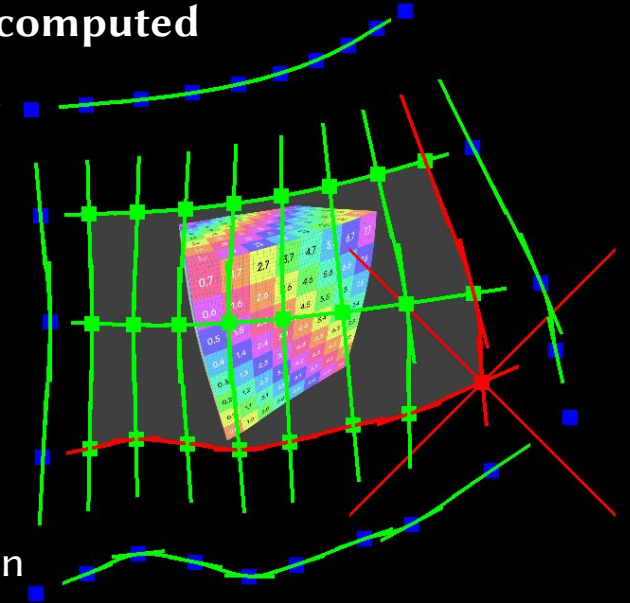
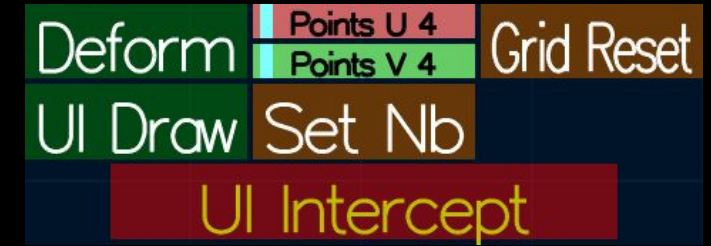
u/v → Constraint the movement in the u(horizontal)/v(vertical) direction

f → (f)ree: remove the u and v direction constraints

Alt u/U → Add/Remove one point in the u direction

Alt v/V → Add/Remove one point in the v direction

- the `c_bdd_grid_adjustable` have more options than the MEU's UI:  
symmetry and type of deformation in particular





# MEU Out Fuzzy edges

- **Definition of fuzzy edges (blended edges)**  
used to have **seamless result** when forming **an image with several projectors, fading nicely from one image to another**, avoiding:
  - side by side images (nearly impossible to set perfectly)
  - A brighter area where the projections overlap
- to achieve a seamless projection from two projections (a tutorial will be done on this)
  - Use deformation grids to set a **perfect geometry overlap**
  - Adjust the **fuzzy edges area on both side of the overlap**
  - **Adjust both gamma at the same time** until the overlap vanish.
- **Fuzzy** → allow edges to be rendered as fuzzy each edge have their own setting
- **Left/Right Bottom/Top** → **Toggle individual side**
- **LR/BT** → set the **edge area affected**
- **Gamma** is applied to transform the edge transition. But there is a mode selector



**Do** → Gamma slider value is applied

**Set** → Gamma slider value is applied and it is used to set a global gamma value with the chosen index. This global gamma is shared by all meus Out.

**Get** → get the Gamma value from a global gamma value with the chosen index, then this value is applied. This global gamma is shared by all meus Out.

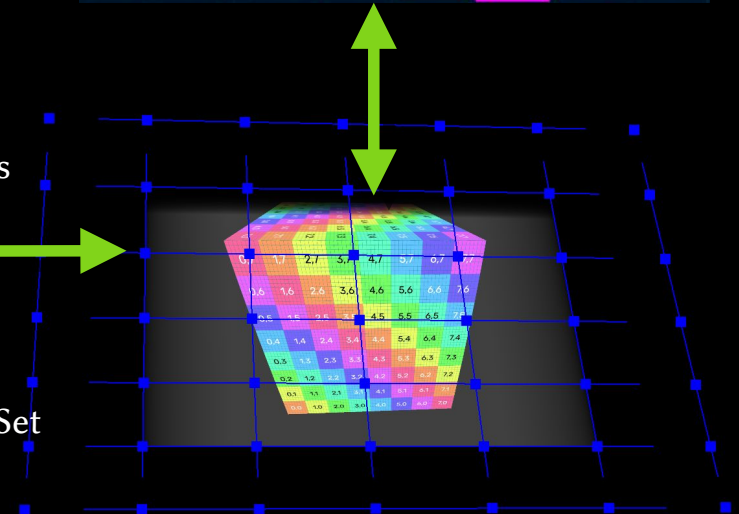
- **In our example**

Fuzzy is done for Left, Right and Top edges.

for **Left Gamma** is 0.339 and this value **set global Gamma slot 1**

for **Right Gamma** is read from global Gamma slot 1, and so the value used here is 0.339 (Set by Left)

for **Top Gamma** used is 0.924 but it does not set any slot



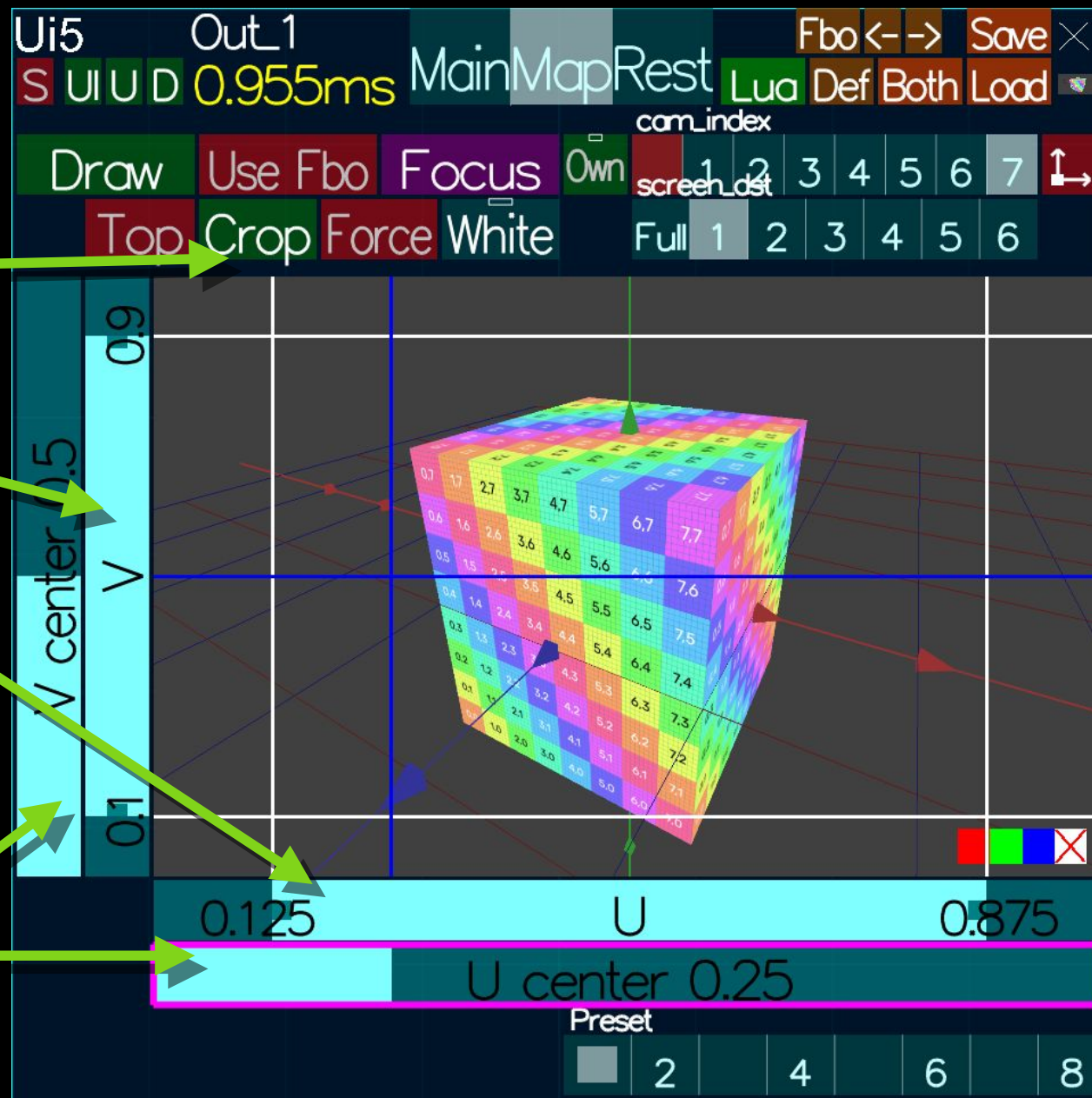
# MEU Out Map tab

- Select which part of the texture is displayed

- when Crop button is On

U/V SLIDER\_TWO selects the area cropped

- U/V center just display the corresponding blue line on top of this texture monitor.



# Rendering BUs (used in other Meu type)

- Control some **common** rendering parameters

Every 3d object is drawn basically with Triangles.  
Each triangle is oriented and face (Front) or not (back) the camera.  
We can draw

Back faces  
All faces  
Front faces

When applicable  
Line width (in Pixel)

Some 3d object have  
Normal to triangle, edge  
or points.  
Switch the Normal  
Display

Do we use Light ?  
Don't for now until you know  
how it works



Factor multiplying the  
Normal Size to make  
more or less visible

Do we use Depth (Zbuffer) ?  
We need the Fbo to have a  
Depth attachment

Point size (in Pixel)  
When applicable

Front Faces and Back Faces can be drawn using different modes.  
Fill (plain) or Line or Point  
These selector pick this mode

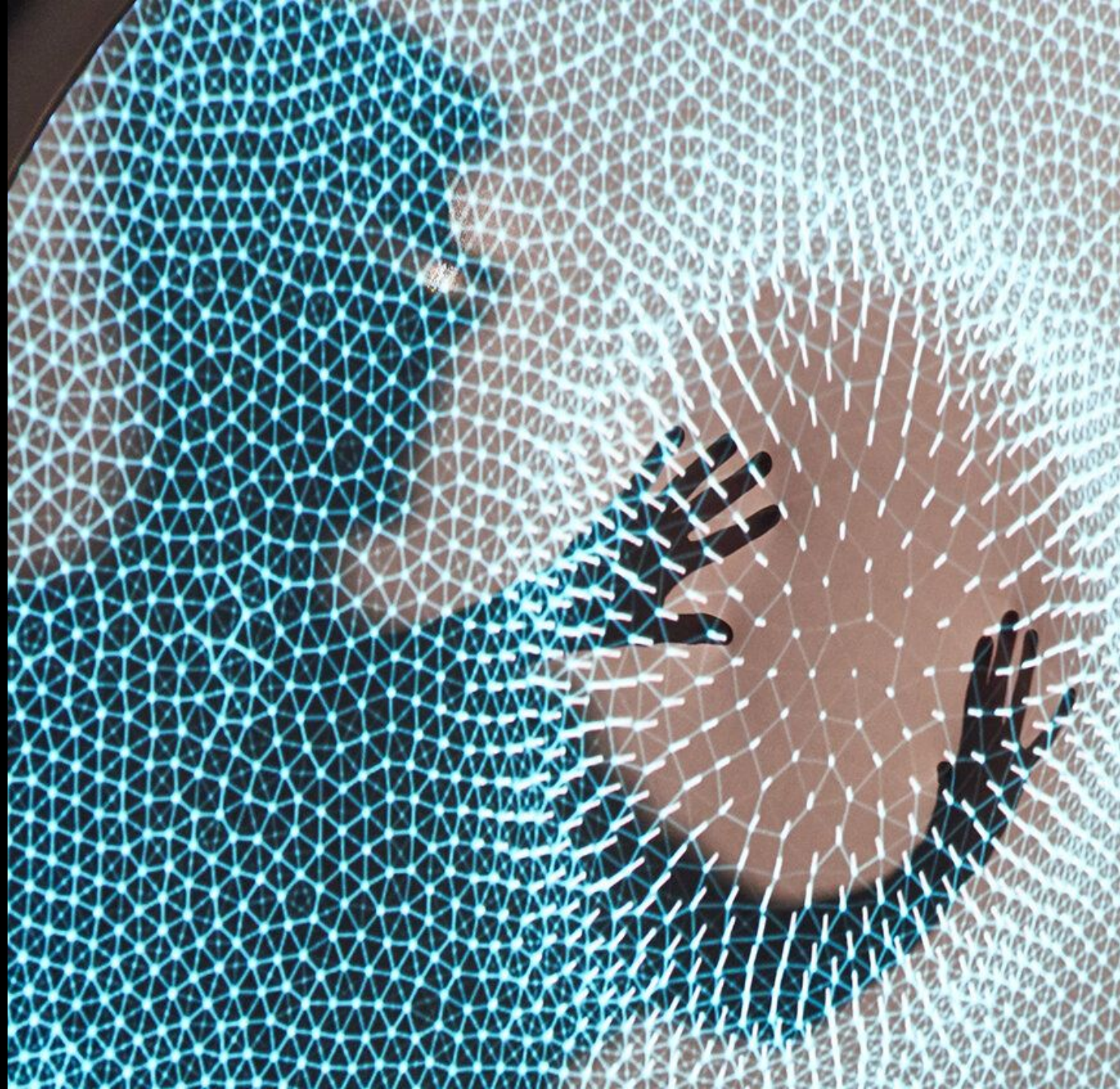


# AAASeed

## An introduction

### Part 22: Plugging

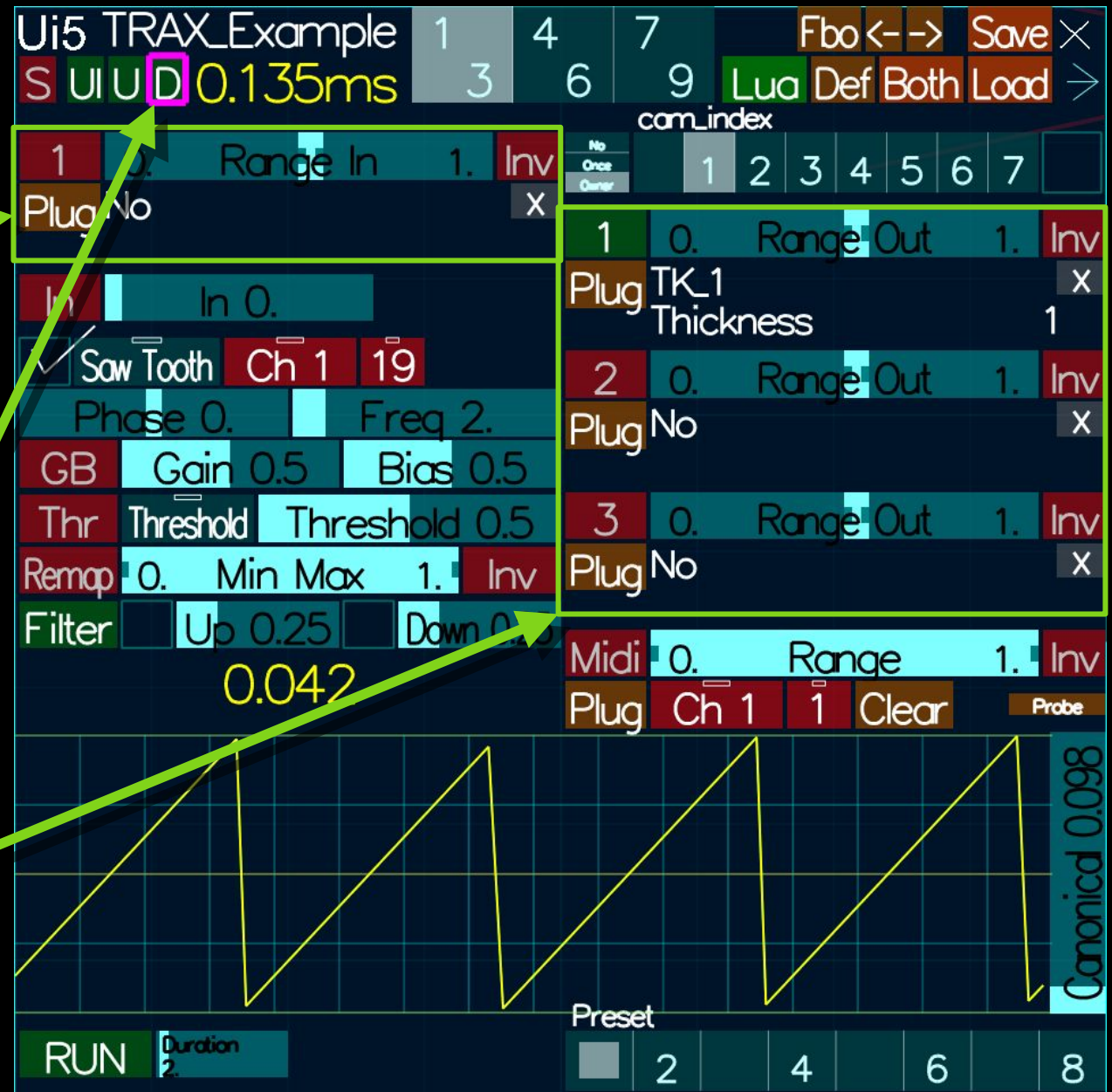
- MEU Trax function
- Plug 101
- MEU Trax UI





# MEU Trax

- It gets a signal/value from
  - any BU in any MEU
  - Plug In MEU Trax
  - MIDI (controller mainly)
  - mouse or keyboard
  - Sound input
  - Internal generators
    - sinus, saw tooth, triangle, square, random, turbulence....
- Process it
- Displays it in its UI
- Draws it to current Fbo
  - Set D button Off in meu bar to avoid
- Propagates it to
  - any BU in Any MEU
  - Plug Out MEU Trax
  - a AAASeed internal MIDI Controller



# MEU Trax Plug 101

- To plug In or Out to a BU (in any MEU)

1/ you need to choose your UI element

Plug In Top Left

Plug Out top Right

3 elements for each tab

9 possibles Plug Out

2/ push the Plug Button

which became bright Green

3/ push any BU from a MEU

including MU slider

the interface show where it is Plugged

Meu name, Bu Name and values index (some BU have several values)

these field can also be directly edited

4/ Activate the element

5/ Set the Range Out

to remap the Trax value [0,1] interval

Inv button just inverse the Remap

1	0.	Range	Out	1.	Inv
Plug	No				X

1	0.	Range	Out	1.	Inv
Plug	No				X

1	0.	Range	Out	1.	Inv
Plug	TK_1				X
	Thickness				1

1	0.	Range	Out	1.	Inv
Plug	TK_1				X
	Thickness				1

- To unplug

plug to another BU will implicitly unplug

use the X button on the left

- BU plug to an active trax bear a flashing green triangle

Top Left for Bu Destination (Trax Out)

Bottom right for BU Source (TRAX In)

Thickness 0. Thickness 0. Thickness 0.

# MEUTRAX Ui

- Select the input or internal Generators



- Process The Signal

GB Gain Bias (see slide for Meu Video)

Threshold

Mode Flip On Up useful to  
transform push button in switch

Remap

Filter going Up andor Down



- Display the signal

Cyan input signal

Yellow processed signal

- Control Signal display

