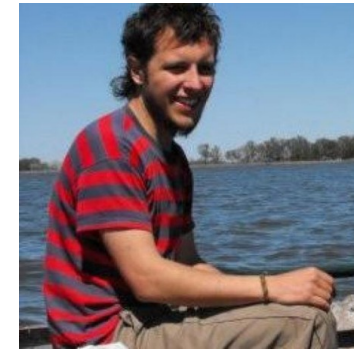


PharoSDL3

A New Window into Pharo's Graphics Stack

Martín Dias & Pablo Tesone | ESUG 2026, Plovdiv



Martín Dias



Pablo Tesone



About the Presenters



Martín Dias

- Pharo Consortium Engineer
- Email: tinchodias@gmail.com
- GitHub: @tinchodias



Pablo Tesone

- Pharo Consortium Engineer
- Email: pablo.tesone@inria.fr
- GitHub: @tesonep



The Transition to SDL3

- SDL2 Deprecation: The industry is moving to SDL3
- Why SDL3?
 - Same cross-platform support
 - Rich desktop integration
 - Heavy GPU usage
- Ready for Pharo 14 release





What is PharoSDL3?

Everything included in Pharo 14

- Complete Bindings: Complete FFI coverage of the SDL 3.0 dynamic library
- High-level and low-level abstractions
 - Direct FFI Access: Developers can bypass abstractions and talk directly to C API
 - Backward Compatibility: Fully supports generic OSWindow driver API
- Included in latest Pharo 14 VM



Rich Desktop Integration Features

- Parent/Child Windows: Utility, Tooltip, and Modal window relationships
- Native System Tray menus
- Native open file dialogs
- Clipboard with MIME support
- Logical display layout
- Touchpad finger tracking and pinch gestures
- Audio playback and capture
- Capture camera

TDLR; Watch demos

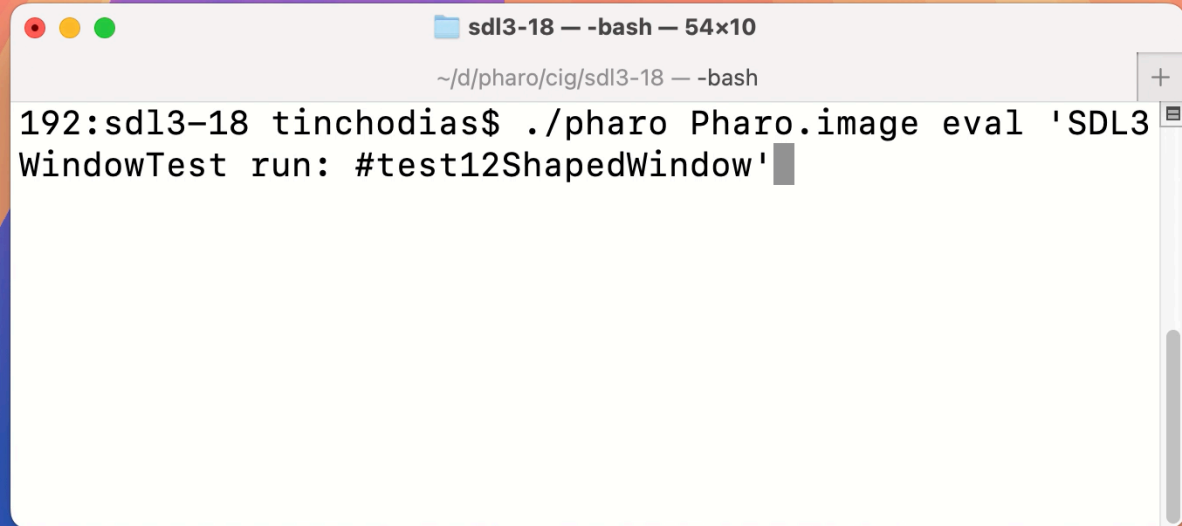
Demo time!



SDL3TrayMenuDemo - ESC to quit

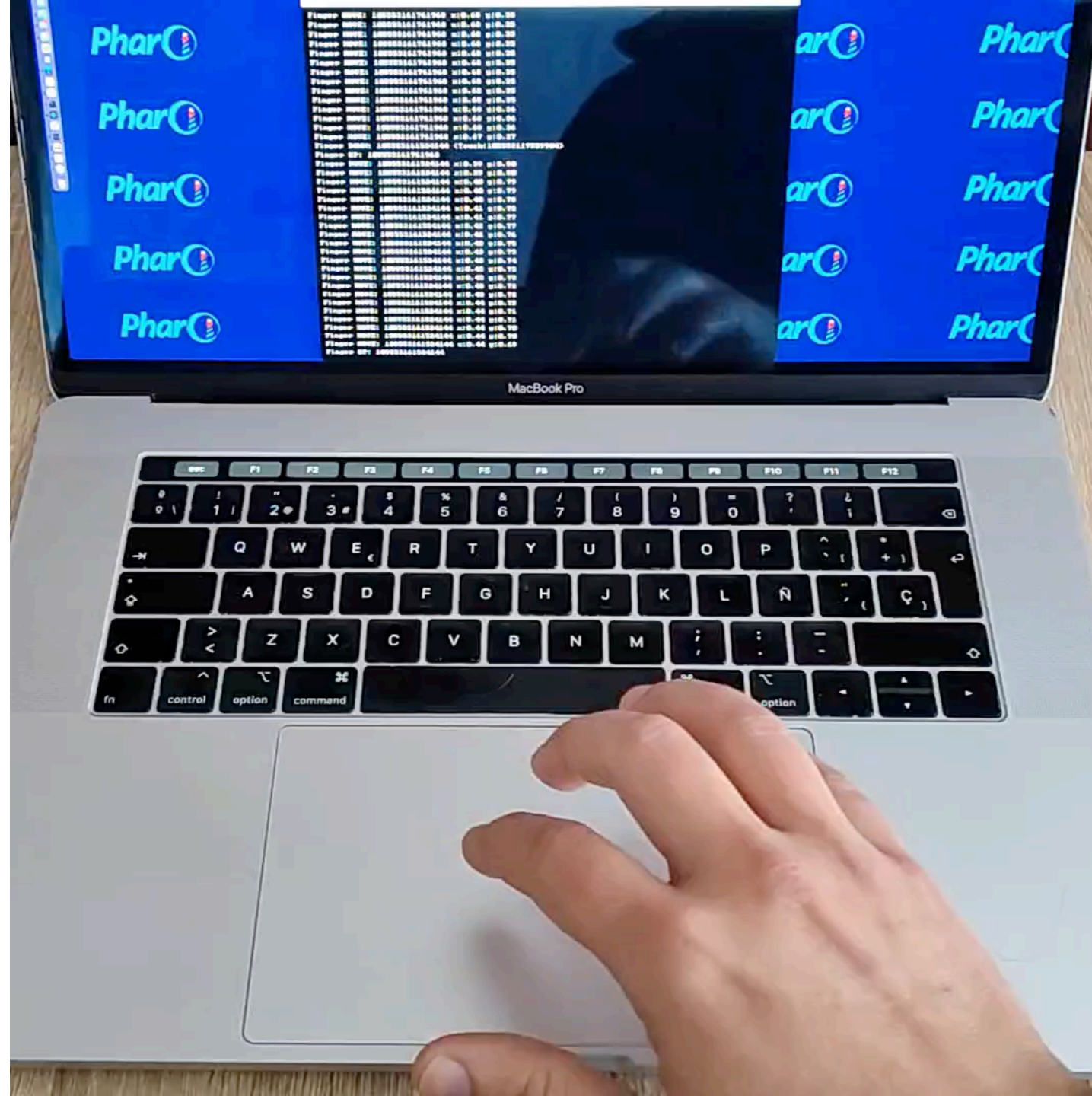
Click on the system tray menu with Pharo icon.
Click on 'Button Two'
Click on 'Checkbox Five'

Shaped Window

A terminal window titled "sdl3-18 — -bash — 54x10" is shown against a colorful, abstract background. The window's title bar includes standard macOS window controls (red, yellow, green buttons) and a close button (+). The terminal content shows a user prompt "192:sdl3-18 tinchodias\$" followed by the command `./pharo Pharo.image eval 'SDL3 WindowTest run: #test12ShapedWindow'`. The cursor is positioned at the end of the command line.

```
sdl3-18 — -bash — 54x10
~/d/pharo/cig/sdl3-18 — -bash
192:sdl3-18 tinchodias$ ./pharo Pharo.image eval 'SDL3
WindowTest run: #test12ShapedWindow'
```

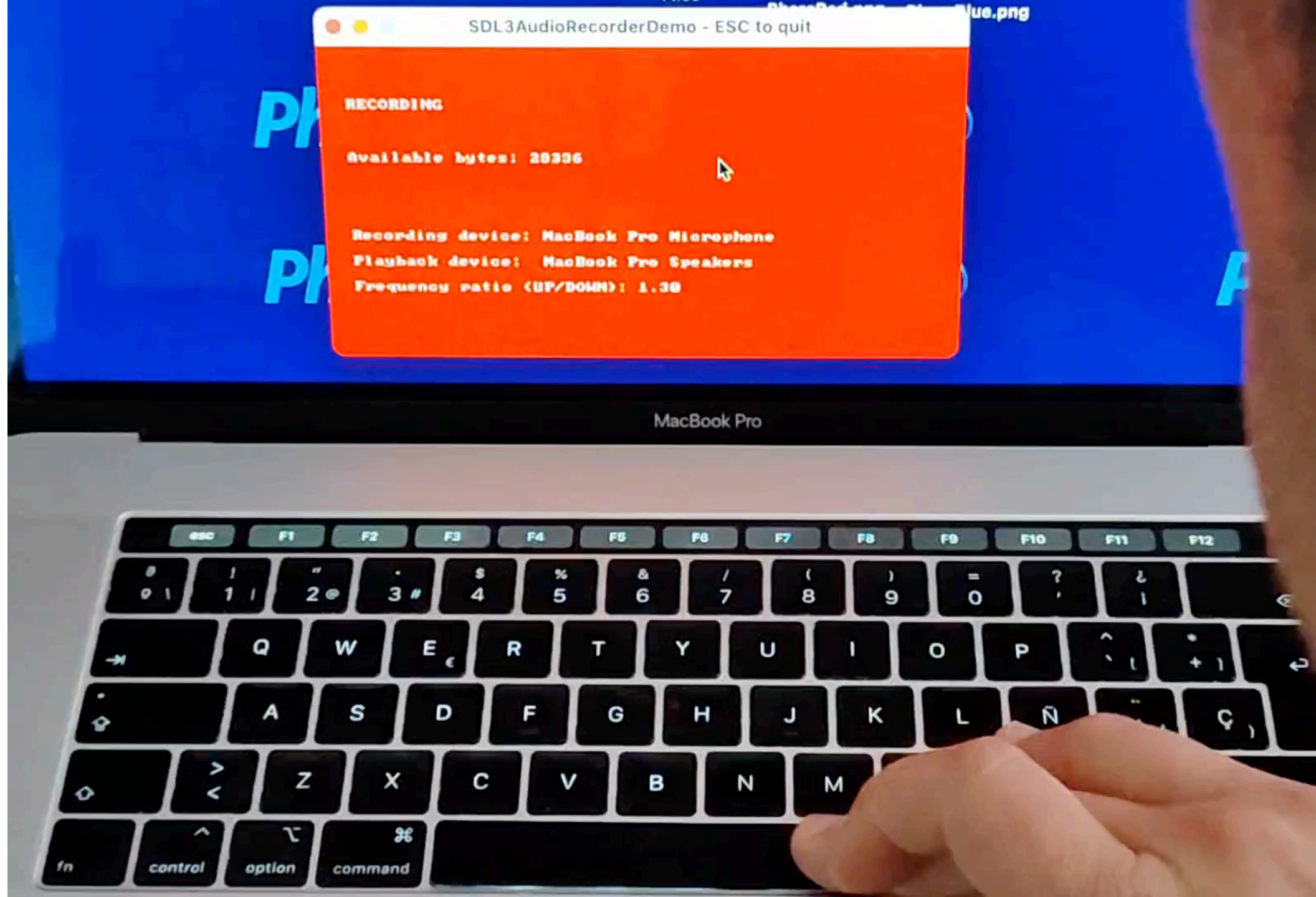
Fingers



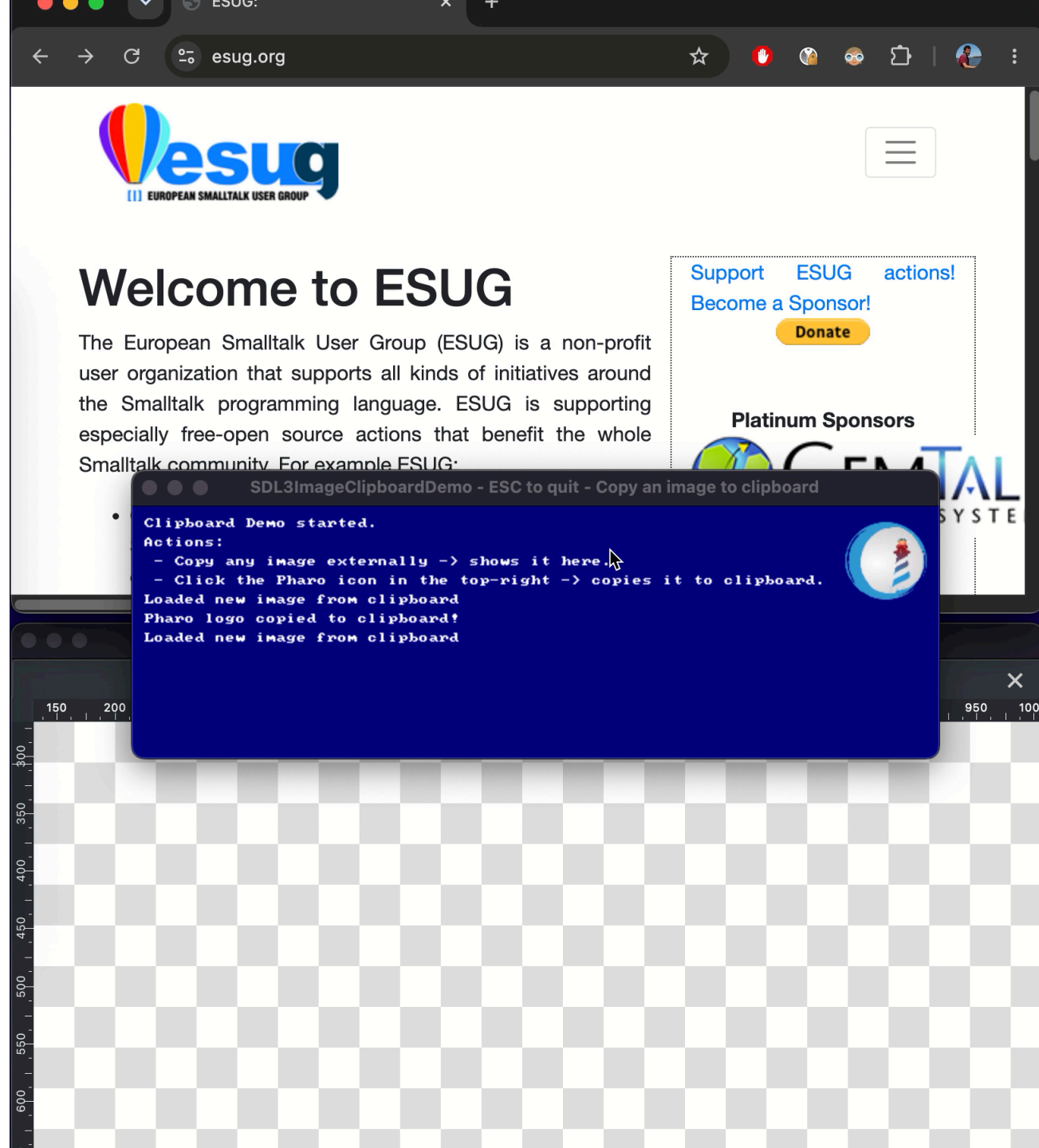
Pinch to Zoom



Audio: Record & Play

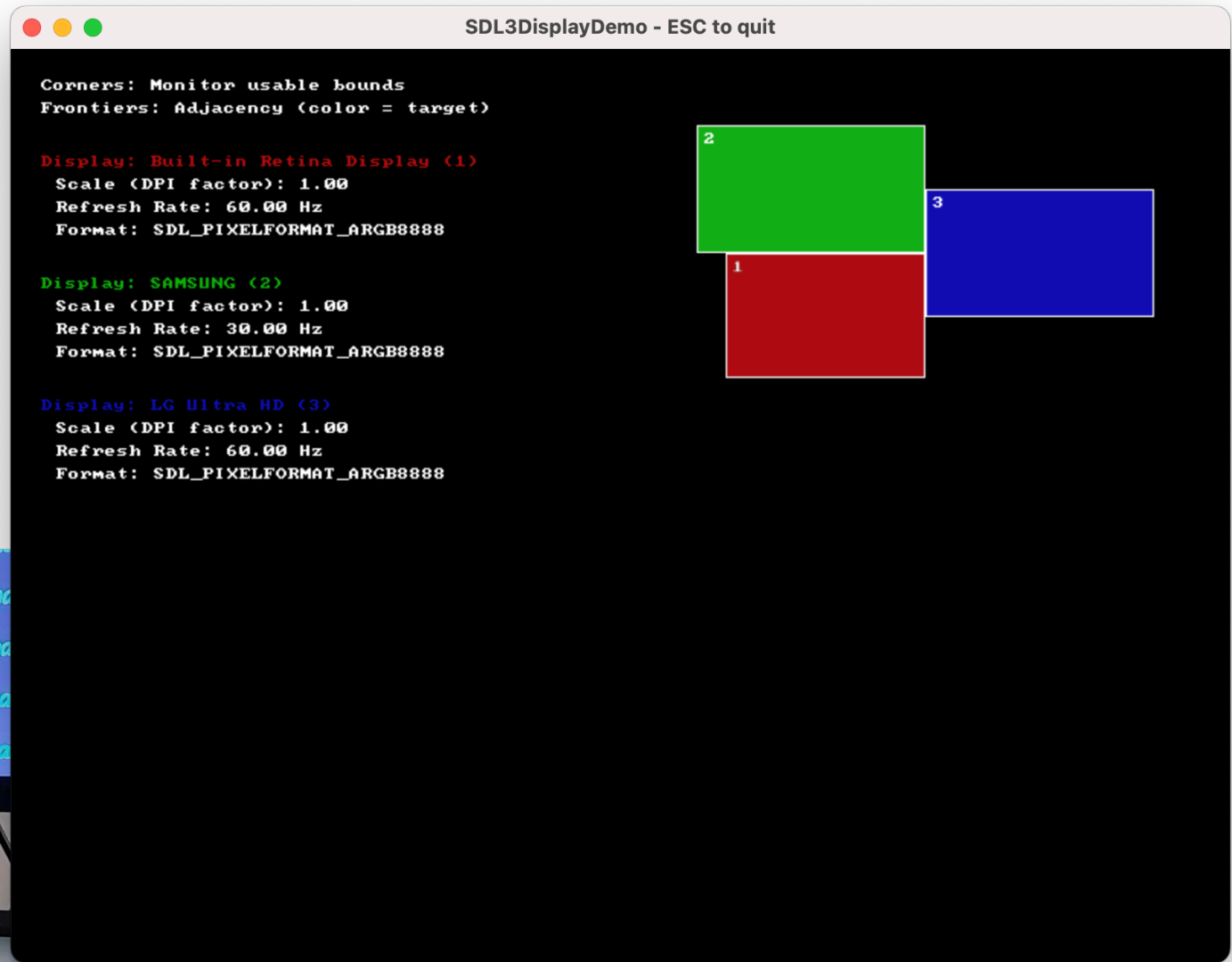


MIME Clipboard: Copy & Paste

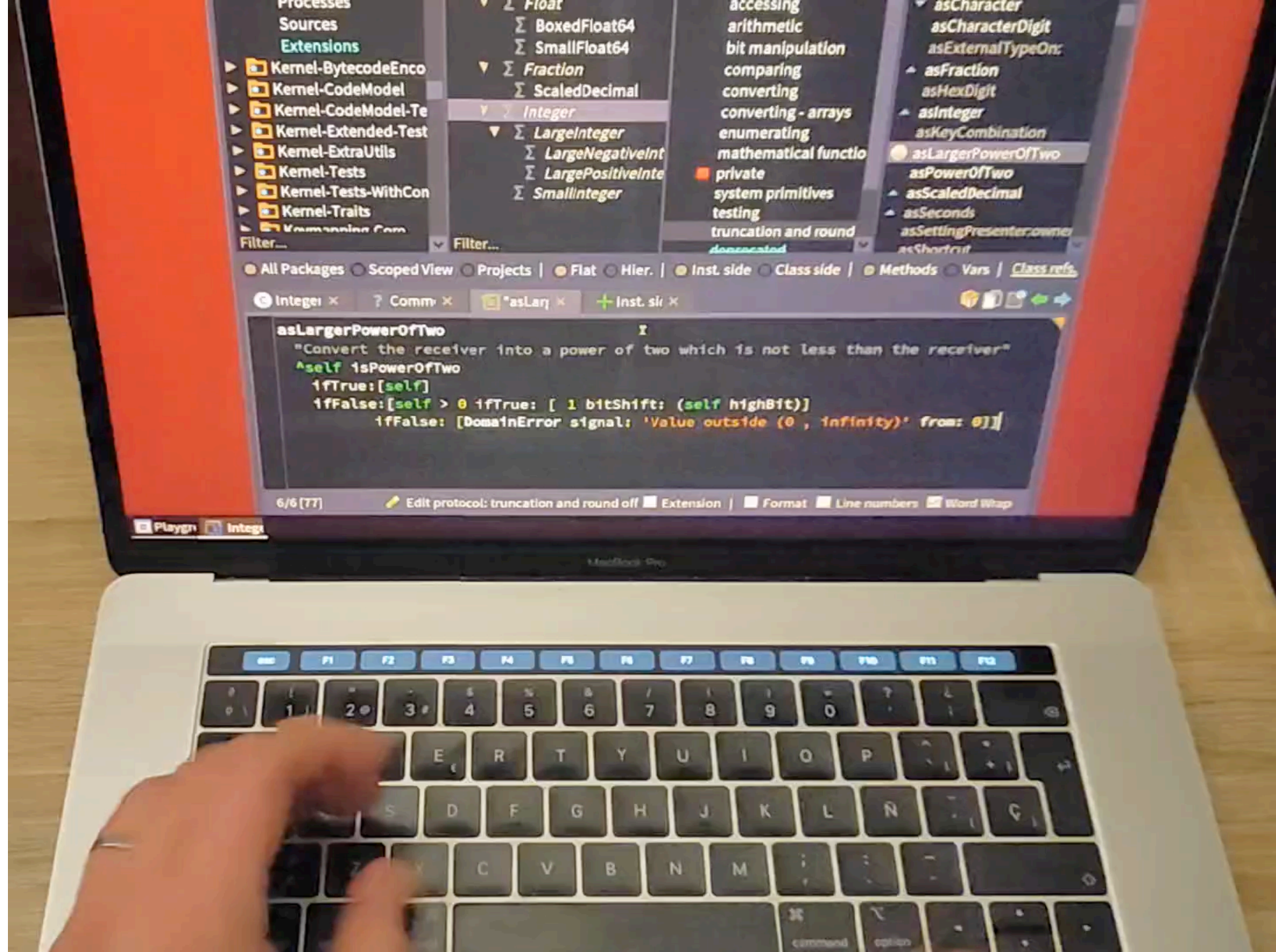




Displays Information



Our Patch to SDL3





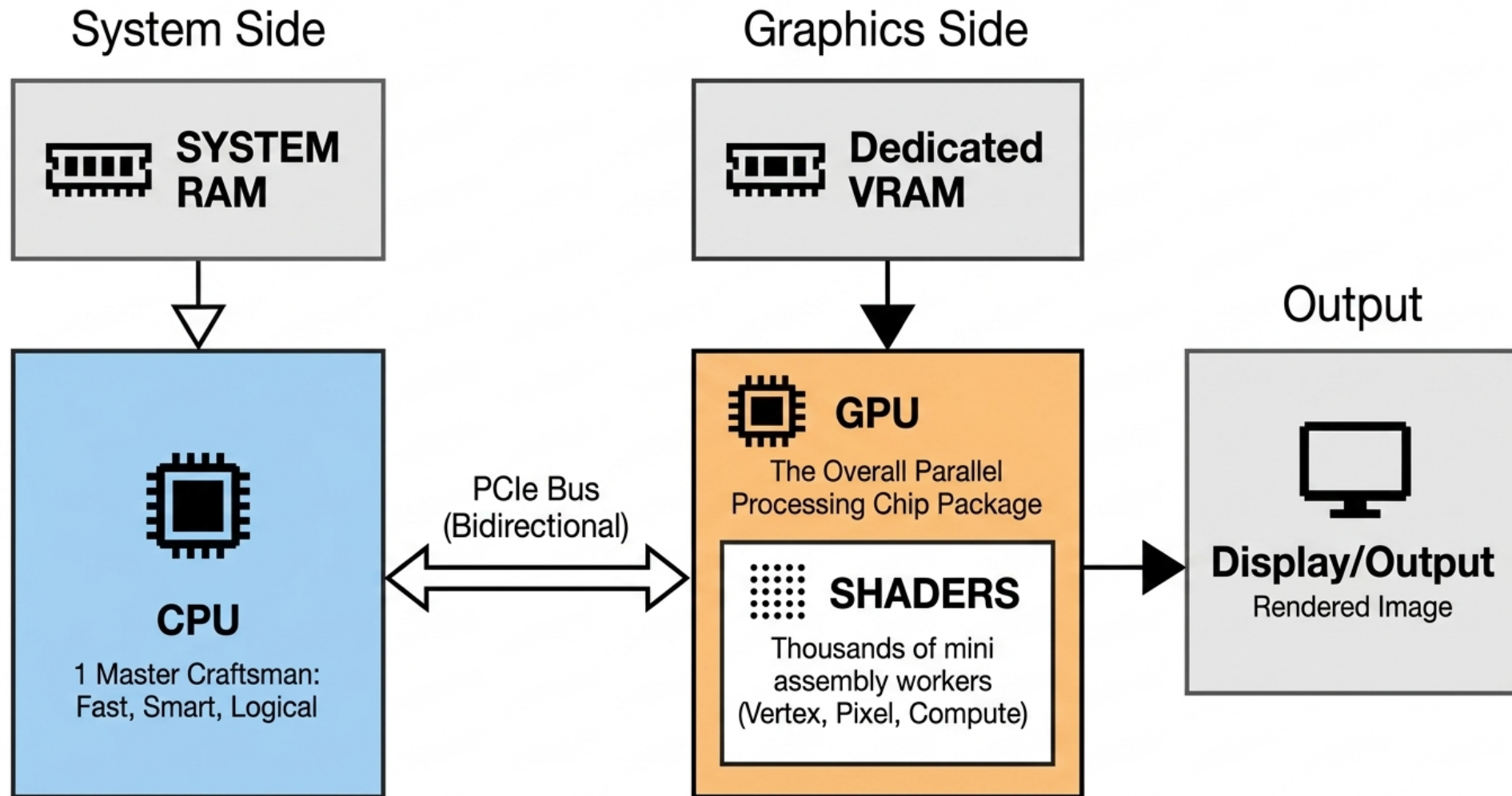
API Completeness, Tests and Demos

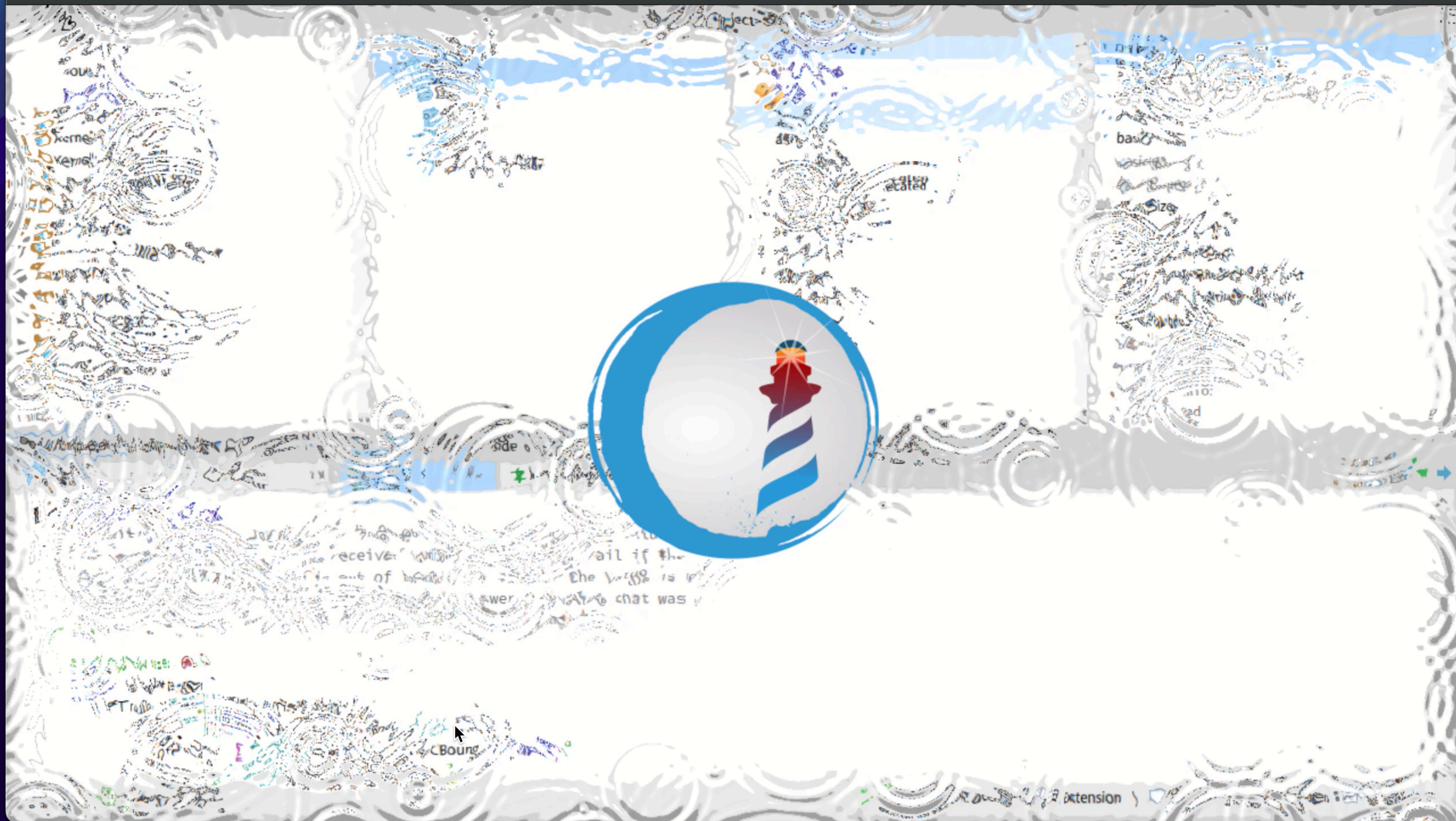
- Bootstrap Importing: C header parsing driven by Esteban's CIG (github.com/pharo-cig/pharo-cig)
- Naming conventions to ease mapping C code to Pharo code
Example:
 - `SDL_CreateWindow(title, w, h, flags)`
 - `LibSDL3 >> #createWindowTitle:w:h:flags:`
 - `LibSDL3 >> #newWindowTitle:w:h:flags:`
 - `SDL3Window class >> #unsafeNewTitle:w:h:flags:`
- Good test and demo coverage for key features

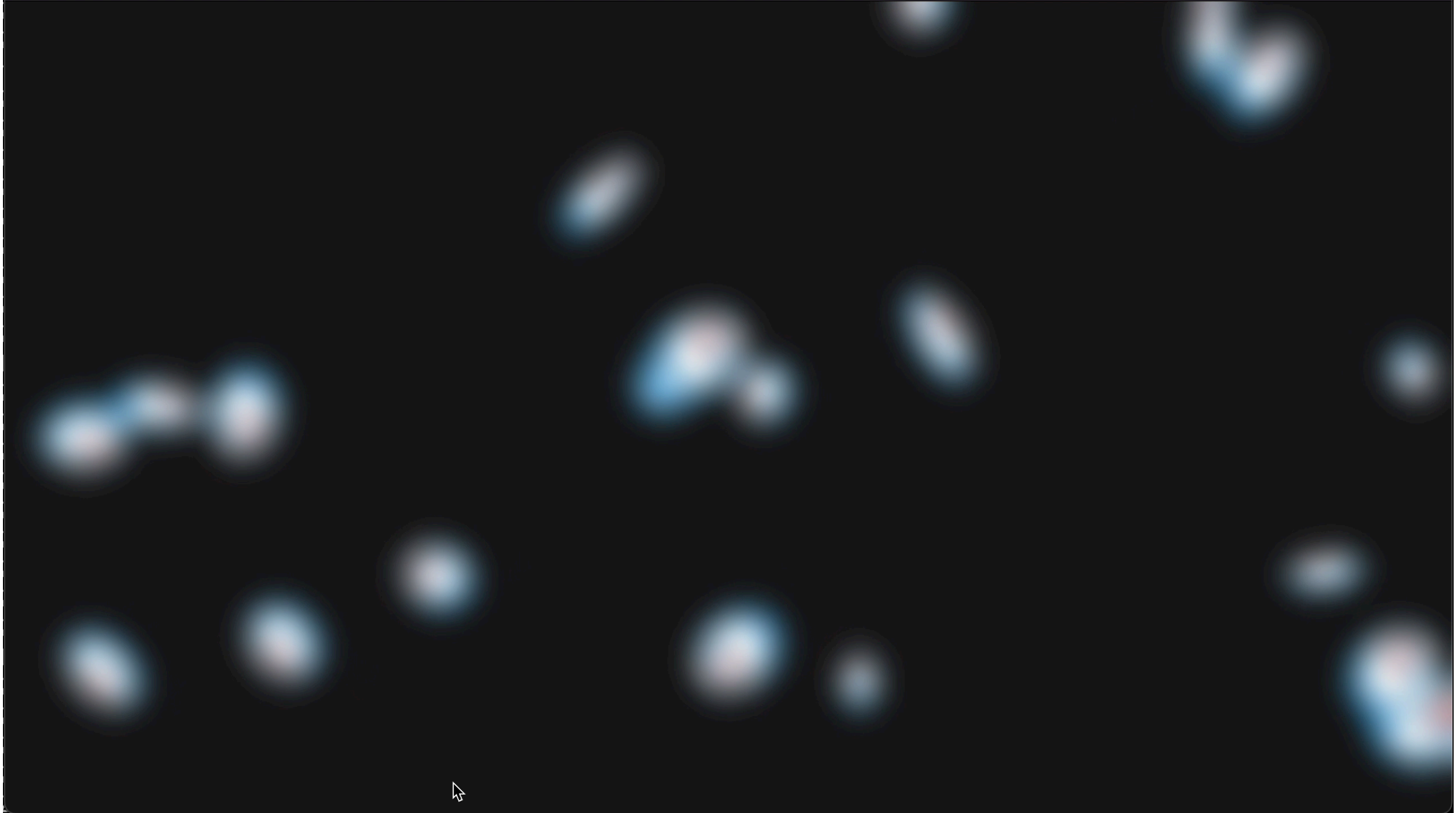
SDL3 GPU Pipeline



CPU vs. GPU Memory Model





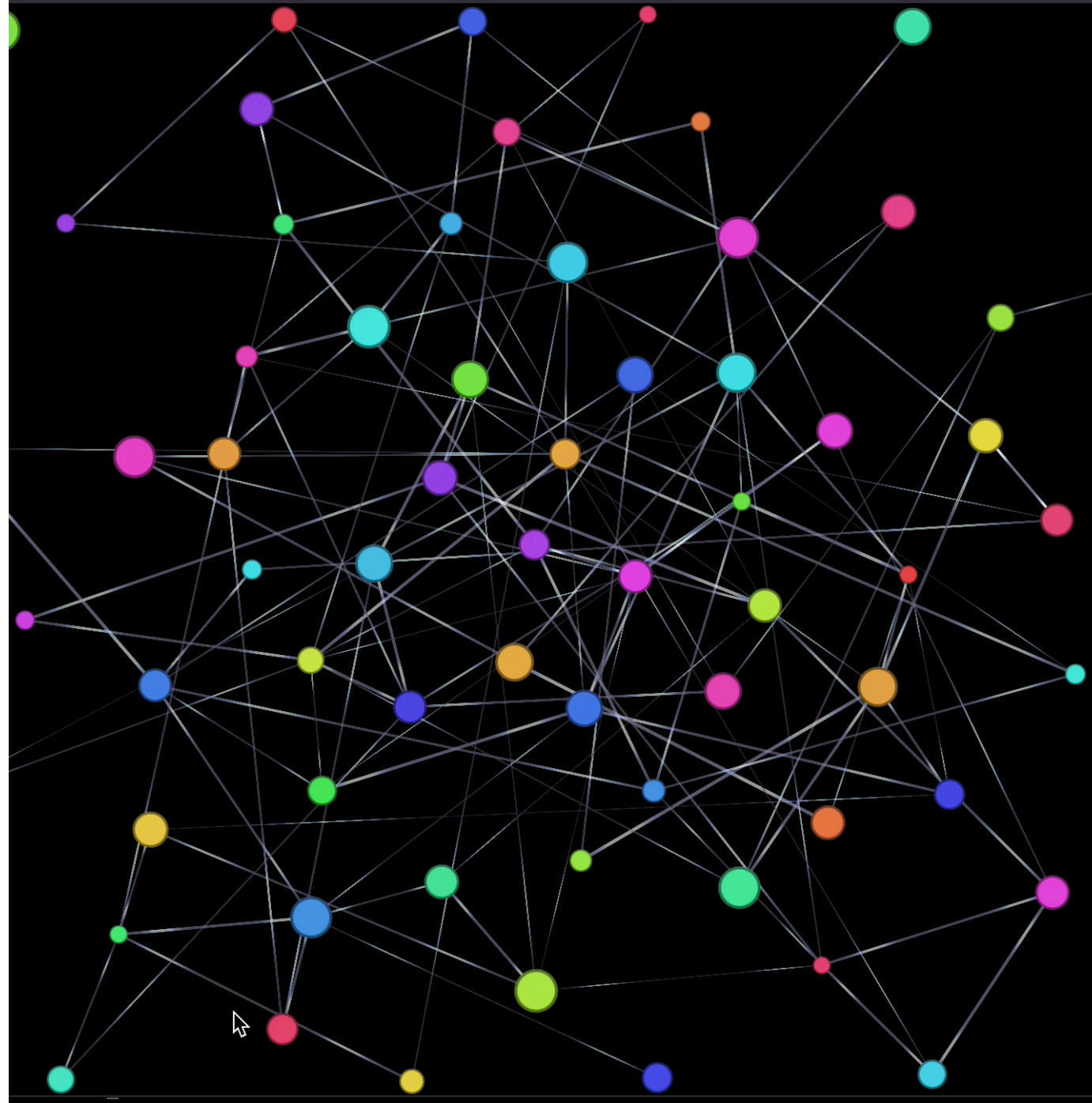


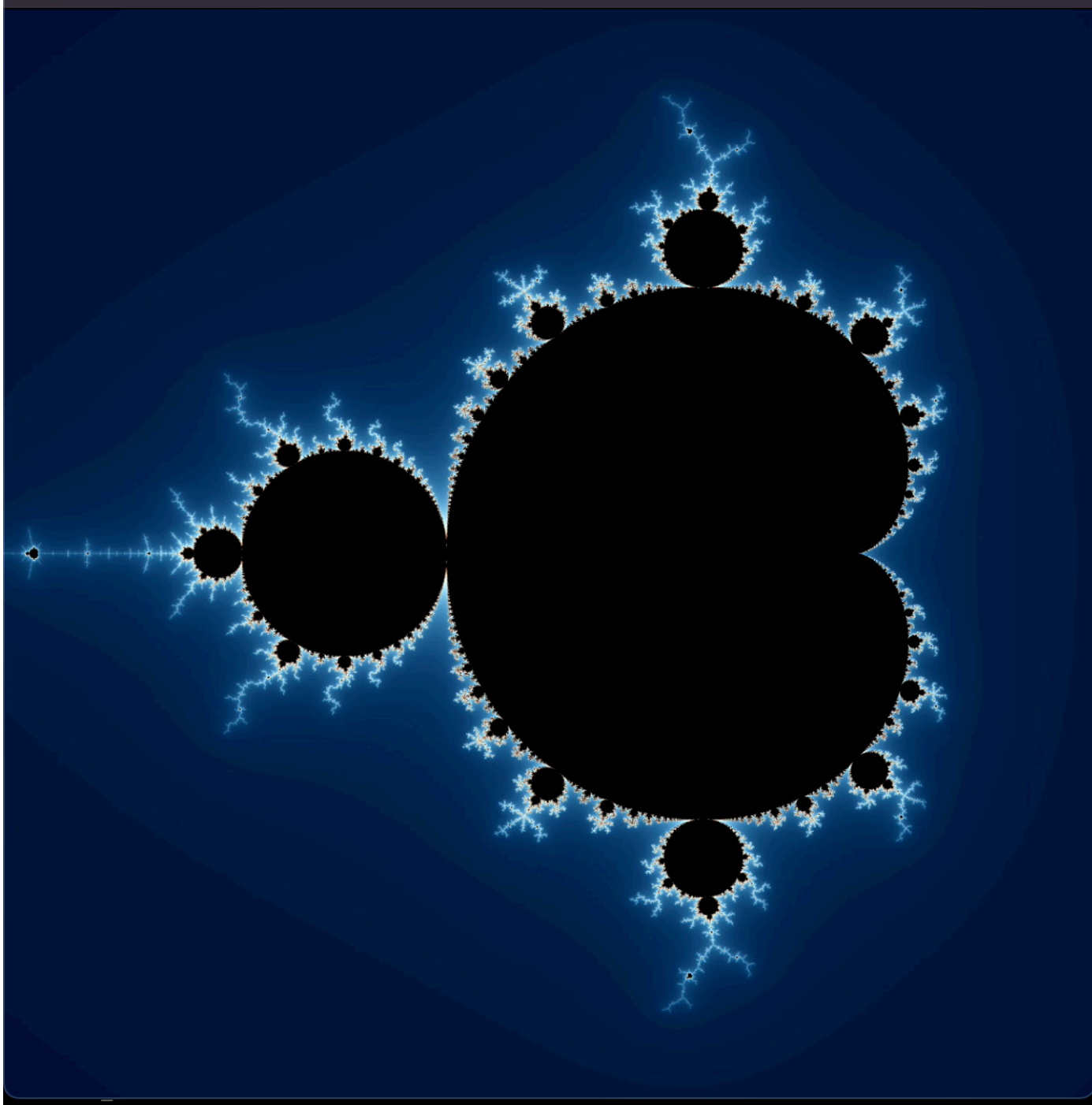


SDL3GPUBoidsDemo - ESC to quit - Count: 1000 | Sep: 1.5 Ali: 1.0 Coh: 1.0 | Rad: 0.05/0.20 | MaxV: 2.00

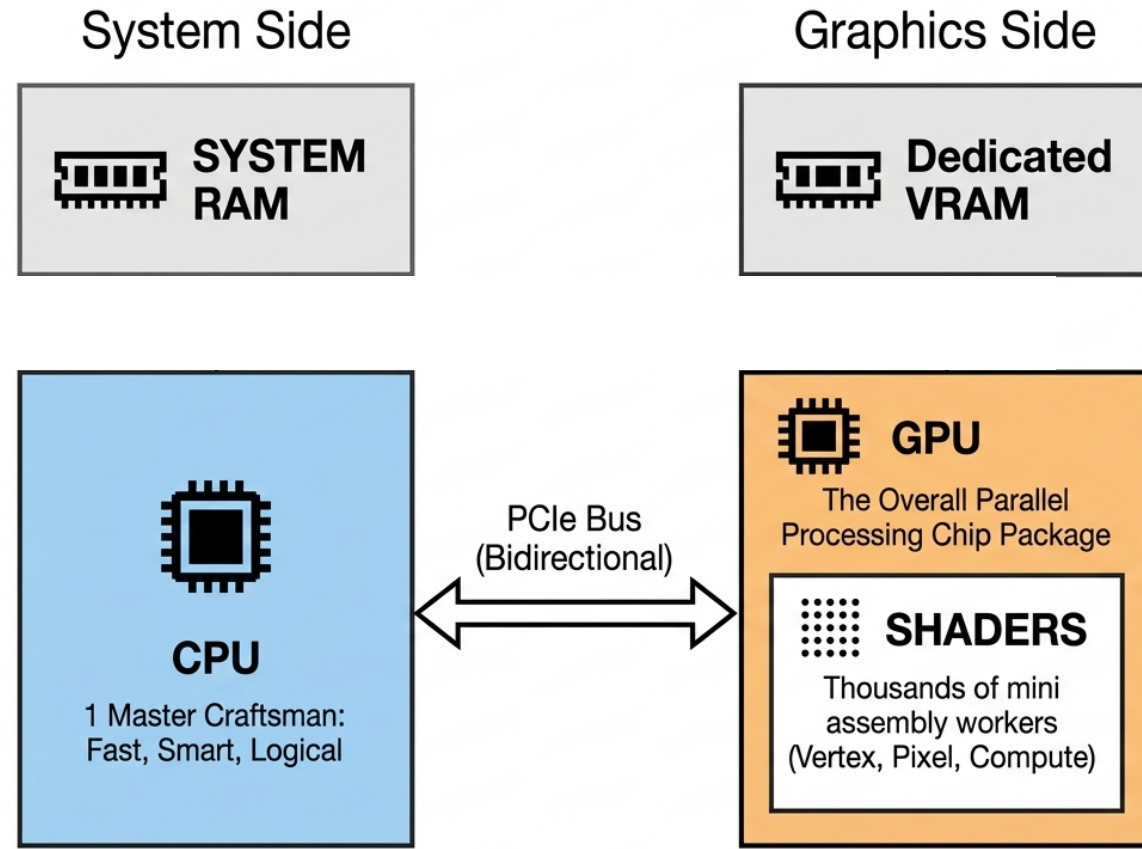


SDL3GPUNodeForceDemo - ESC to quit - Nodes: 64 Edges: 128 (Spring: 2.0 Repel: 6.0 Damp: 0.0100)



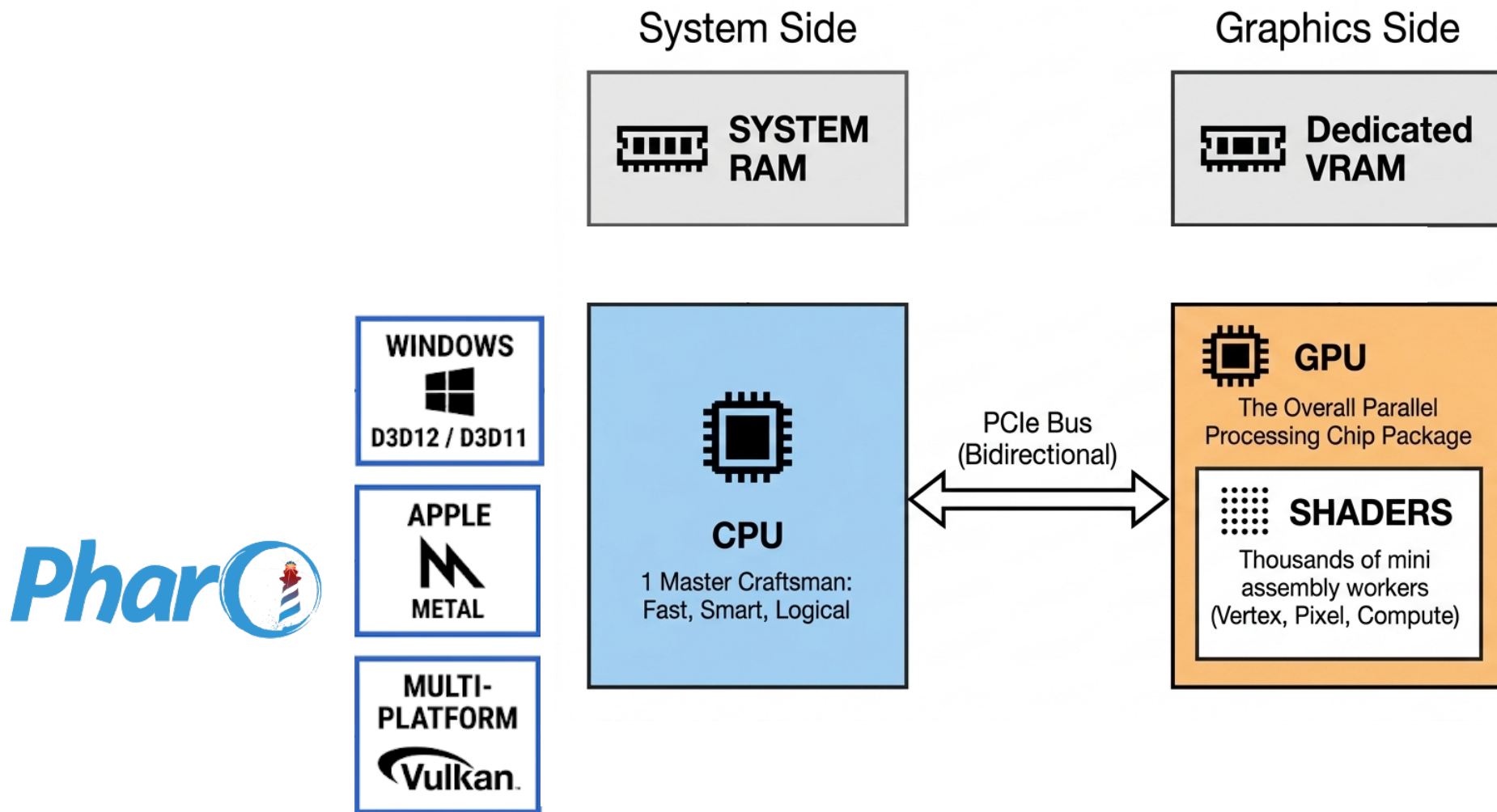


Cross-Platform Support



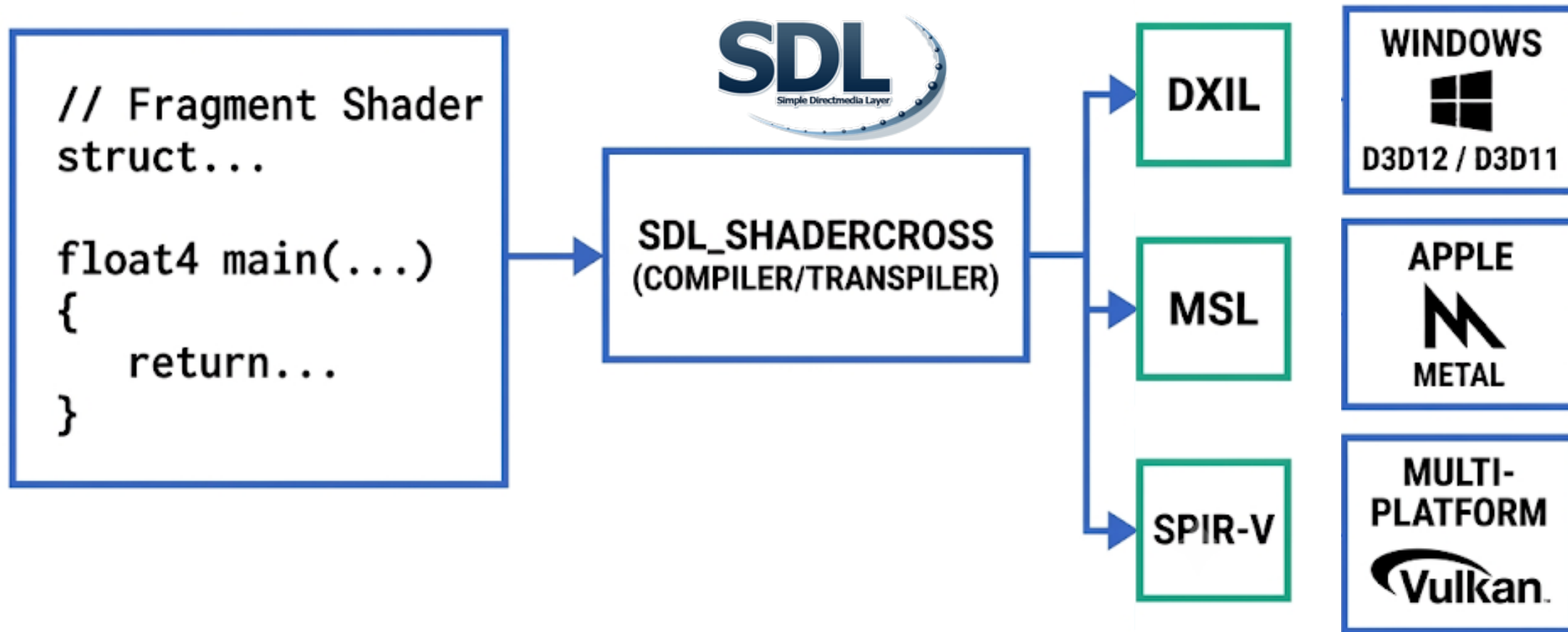


Cross-Platform Support





Cross-Platform Support



Cross-Platform Support



SDL
Simple Directmedia Layer

**SDL_SHADERCROSS
(COMPILER/TRANSPILER)**

Method: SDL3GPUDualKawaseUpsampleFragmentShaderSource class>>hlslSource

class side
accessing
overrides

base64DXILCode
base64MSLCode
base64SPVCode
hlslSource
metadata

Texture2D tex : register(t0, space2);
SamplerState sam : register(s0, space2);

cbuffer BlurUniforms : register(b0, space3) {
float2 texelSize;
float blurRadius;
};

float4 main(float2 uv : TEXCOORD0) : SV_Target {
float2 offset = texelSize * blurRadius;

float4 sum = tex.Sample(sam, uv + float2(-offset.x * 2.0, 0.0));

Method: SDL3GPUDualKawaseUpsampleFragmentShaderSource class>>base64DXILCode

base64DXILCode
base64MSLCode
base64SPVCode
hlslSource
metadata

base64DXILCode
"Generated from #hlslSource"

'RFhCQ6WwKLSDIHUQF/t5i4AzM94BAAAA6BEAAAcAAAA8AAAAATAAAIgAAADEAAAApAEAA0AIAA
D8CAAAU0ZJMAgAAAAAAAAAAAAAAAAELTRzE0AAAAAQAAAgAAAAAAAAAKAAAAAAAAAAAAwAAA
AAAAAADAwAAAAAAAFRFWENPT1JEAAAAAE9TRzE0AAAAAQAAAgAAAAAAAAAKAAAAAAAAABAAAA
AwAAAAAAAAAPAAAAAAAFNW1RhcmdldAAAAAFBTVjJdYAAAAANAAAAAAAAAAAAAAAAAAAAAA
AAAAA/////wAAAAABAQABAQAAAAAAAAAAAAAAAAAAAAoAAAAADAAAGAAAAAIAAADAAAAAAAAA
AAAAANAAAAAAAAAAEAAAACAAAAAAAAAAAAAAAAAAAAAAAMAAAAACAAAAAAAAAAAAAAAACAAAAA
AAAAABAAAAAAVEVYQ099PUKQAbWFpbGAAQAAAAAAQAQAAAAQAQAAAAABAEIAAwIAAAAAAA
AAAAAQBEEMAAAAAPAAADwAAAAAAAAAAAAAAAAAU1RBVDQHAABgAAAAzQEAAERYSUwAAQAAEAAABw
HAABCQ8DeIQwAAMQBAAALgiAAAGAAABMAAAAHgSORQcgESQYQmjmSAYQMjQUIGR4Ei2KAGEUCQp
ILQsQQMhQ4CBhLCjJiiEiQFCBDRoiLABkyQuRIDpARI8RQQVGBjOGD5YoEMUYGURgAAAgAAAAbj
OD/////B0ACqA2E8P/////8DIG0whv/////8fAAmoAEkYAAADAAAAE4JgQiBMCAYAAAAAiSAAAFEA

Conclusion

- Direct C API access makes powerful OS features first-class citizens in Pharo
- PharoSDL3 is READY! Download Pharo 14 and play with demos!
- Next talk: Open Bloc with OSWindow-SDL3

