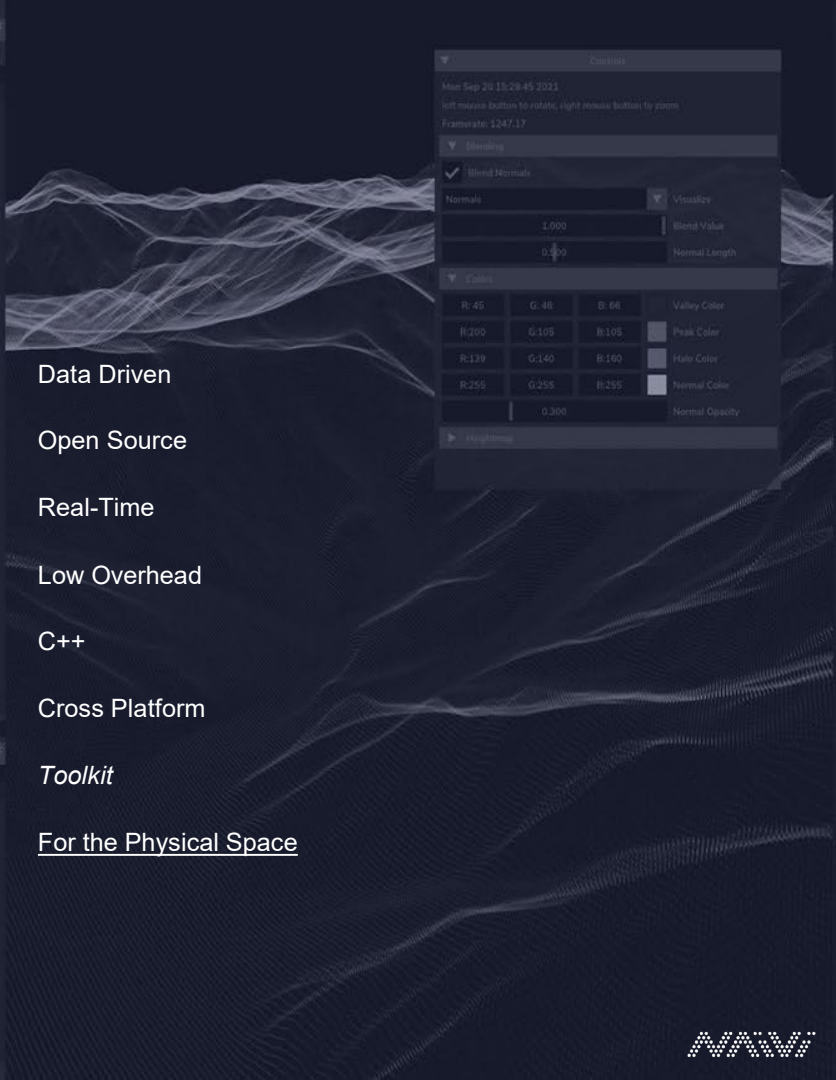
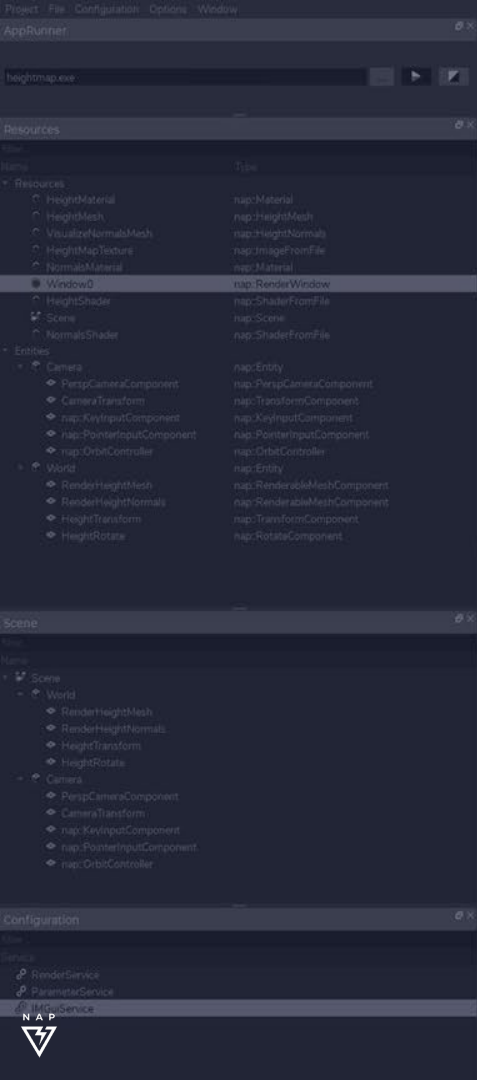




NAP Framework & Vulkan

Efficiently pre-visualize and control physical objects



Data Driven

Open Source

Real-Time

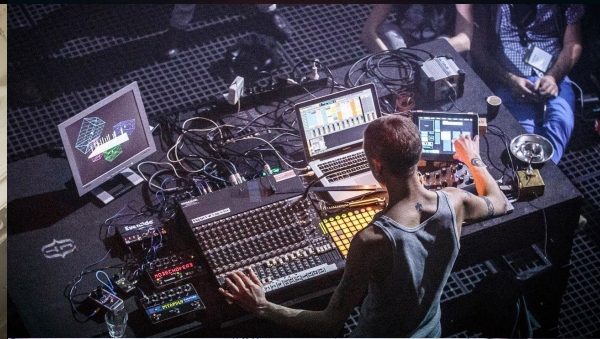
Low Overhead

C++

Cross Platform

Toolkit

For the Physical Space



SHYLIGHT

DRIFT

DRIFT

www



The Problem

old solution

No real-time control over lights

No live sequencing and editing of shows

No live previsualization of complete setup in 3D

Solution did not scale well

Individual cables for Servo & LED

High latency due to inefficient communication protocol

Software was unstable, high maintenance cost



The Goal

new solution

Configure & Sequence complete shows in Real Time

Support 100+ instances

Run on low spec hardware with integrated GPU

With or without the physical installation attached

Accurately pre-visualize movement

Ensure high data throughput for real-time control

DRIFT

NAVI

SequenceEditorGUI

Save As Load Play Pause Goto Loop Follow

100.0 H-Zoom 150.0 V-Zoom

hello world!

00:00:00 00:01:00 00:02:00 00:03:00 00:04:00 00:05:00 00:06:00

Shylight
State: NoCoupler

Vel. Err.
Acc. Err.

Shylight
State: NoCoupler

Vel. Err.
Acc. Err.

Output
Density

00
00

Output
PlayerEventOutput

Music Event Start hangPlayer

Music Event Start hangPlayer2

DRIFT

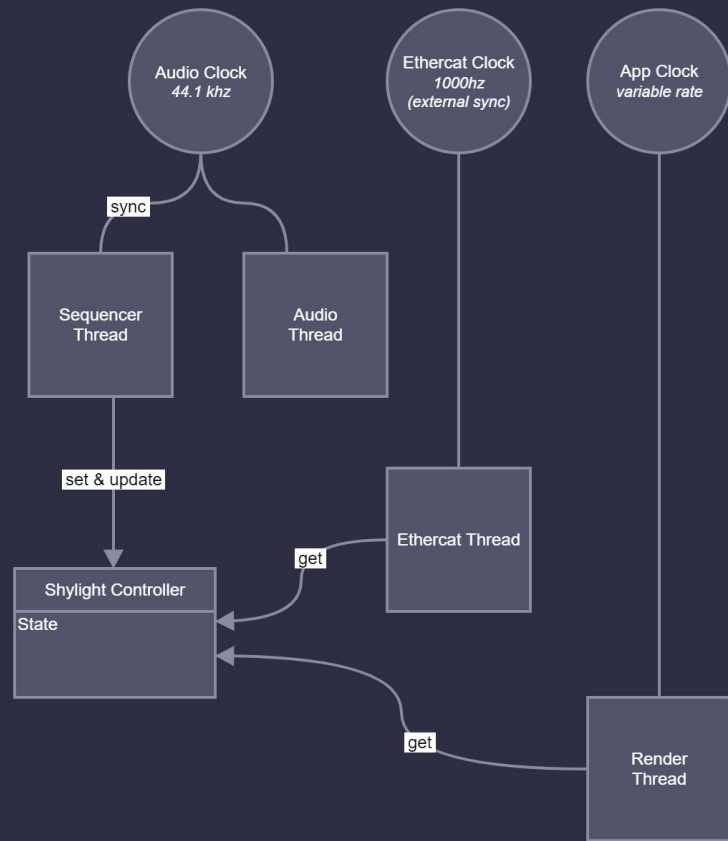
Shylight

Menu Display

Shylight7
Shylight4
Shylight5
Shylight8
Shylight9
Shylight1
Shylight2
Shylight3
Shylight6

Shylight

General Structure



OS / Kernel

Considerations

Windows 10

- 8000 μ s, 125 hz
- Dropped Frames

macOS

- 4000 μ s, 250 hz

Linux

Ubuntu 20.04 LTS Generic Kernel

- 8000 μ s, 125 hz
- Dropped Frames

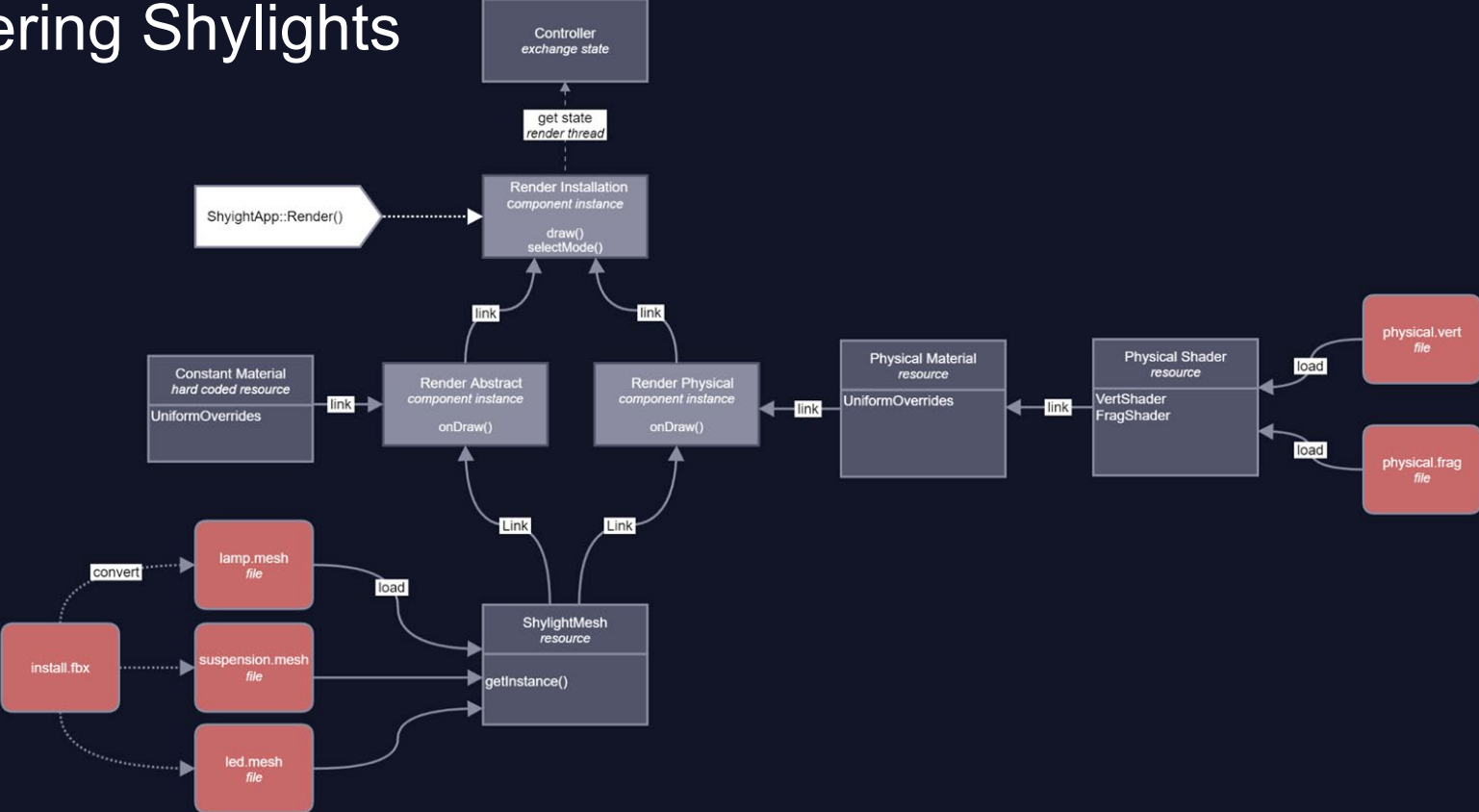
Ubuntu 20.04 LTS Liquorix Kernel

- 1000 μ s, 1000 hz

Realtime Kernel

- 250 μ s, 4000 hz

Rendering Shylights

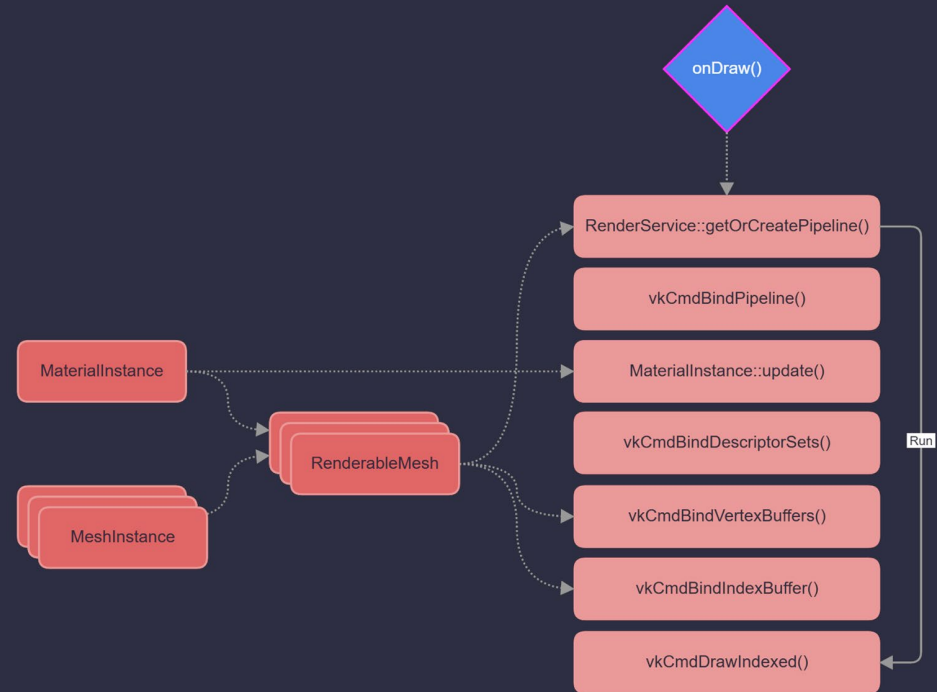
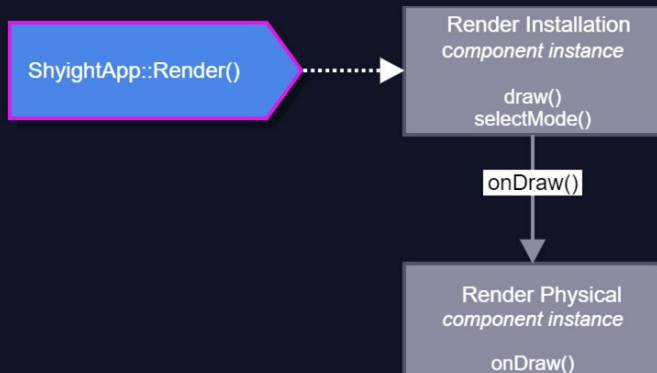


app::Render()

example



nap::RenderableComponent





Conclusion

Key Takeaways

Vulkan simply **works** cross platform

Performance Increase of ~10-40%

Compared to NAP OpenGL backend

MacOS (MoltenVK), Windows & Linux

No noticeable difference for developers

Render interface almost the same

Debug layers are a **godsend**

Headless rendering is now an option

Explicit = Good



WIP

0.5

Official Support for Raspberry Pi

Very promising initial results

Good for pre-visualization purposes

Vulkan Compute

Web Client

Data-driven web-portal

Viewport streaming (0.6+)

Realtime Audio Sequencing

Coen Klosters | Founder & Lead Developer

coen@napframework.com

www.napframework.com

<https://blog.napframework.com/2020/porting-nap-opengl-to-vulkan/>

coen@naivi.nl

www.naivi.nl

Contactweg 47
1014 AN Amsterdam
The Netherlands

N A P



www.napframework.com

open-source software

Special Thanks

Studio Drift

Klaas van der Molen

Jelle van der Beek

Ritesh Oedayrajsingh Varma

Bas Korsmit

Chris Heywood

Tim Groeneboom

Lesley van Hoek

Dodo Kis

Thijs Krooswijk

All of our Users & Clients

The Khronos Group

