
LUNA

An imperative approach to video user experiences using LUNA

William Cooper
informitv

LUNA

Introduction

LUNA engine enables high-performance graphics for video user interfaces.

Alternative to browser-based applications, designed for screen experiences.

Single Application Programming Interface supported across diverse devices and displays.

LUNA

Applications

sky

LUNA

Programme guide

youSee

LUNA Concept

mercury

LUNA Concept

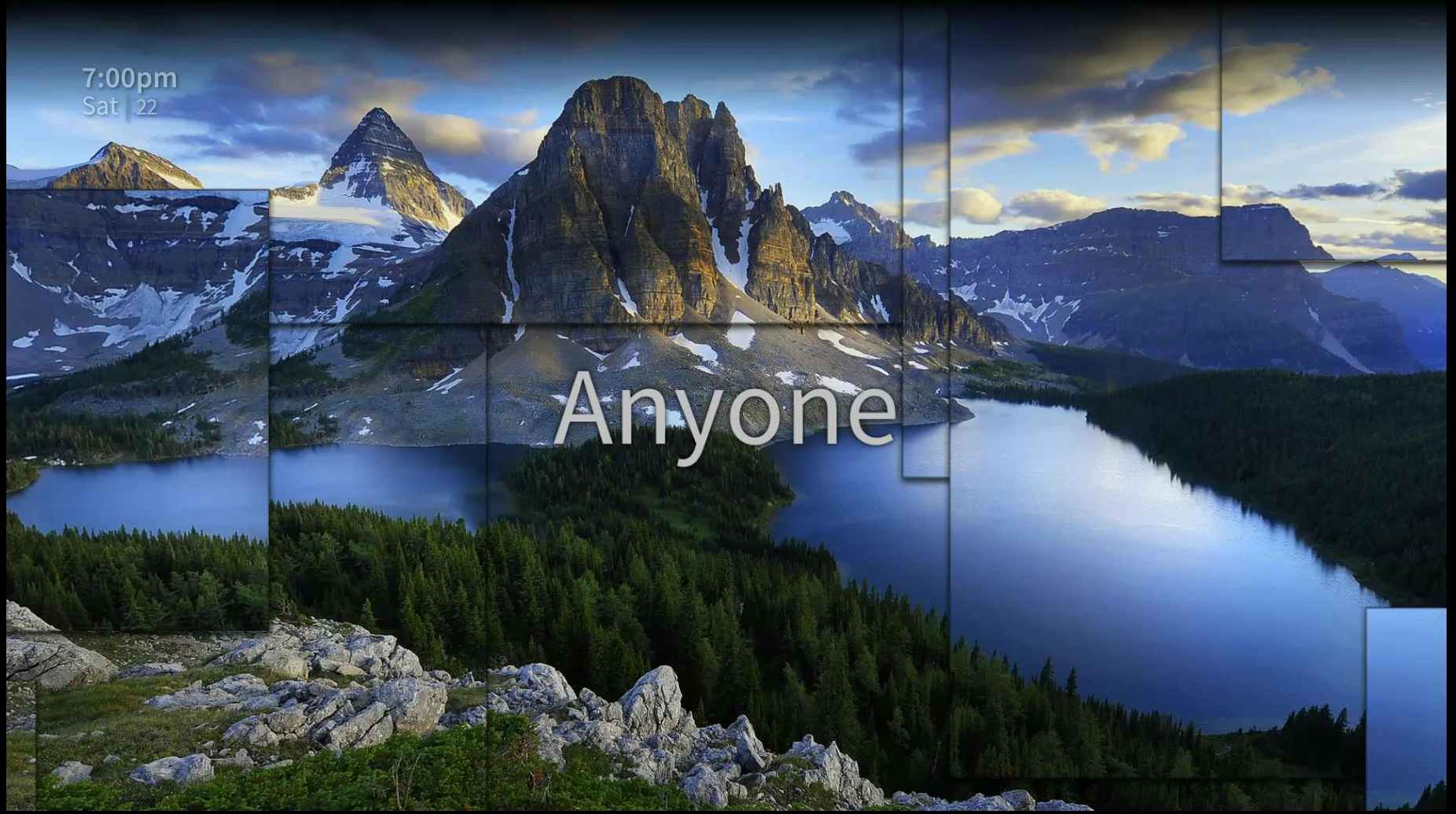
LUNA Concept

Alex

Family

Kids

Charlie


7:00pm
Sat | 22
Anyone

LUNA

Implementation

LUNA

JavaScript+JSON+Images+Video

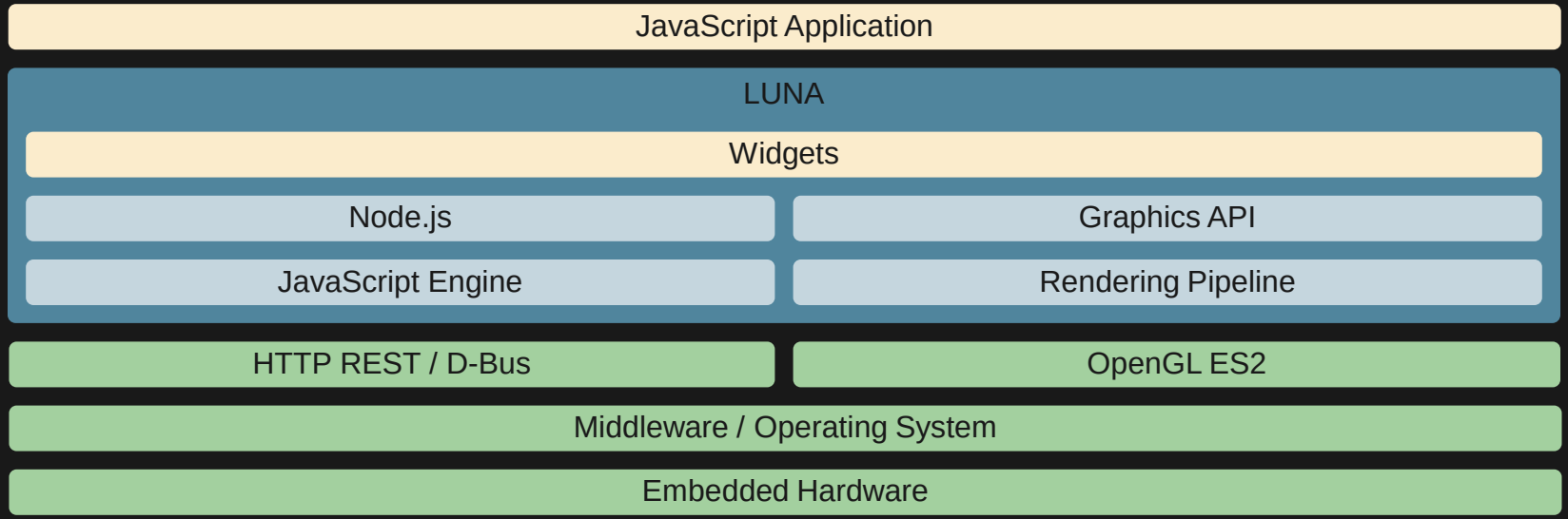
Deployed to Android box as APK app

2 weeks

LUNA Comparison

HTML5	LUNA
Generic	Designed for screen interfaces
Universal	Optimised for performance
Browser	Runtime
HTML+CSS+JavaScript	JavaScript
Document based	Programmatic
Declarative	Imperative

LUNA Architecture



LUNA

Node.js

Designed to run scalable network applications with an event-driven, non-blocking I/O model.

Perfect for modern RESTful application development.

Direct access to file system, web sockets, low-level networking, encryption and compression.

Huge open-source community, with the largest package ecosystem in the world.

LUNA

Graphics engine

Easy-to-use Application Programming Interface based on high-level drawing primitives, designed specifically for video user interface development.

- Fine control of fonts and typography
- High-quality graphics and smooth animation
- 3D objects in 2D scenes, 2D scenes on 3D objects
- Real-time rendering effects
- Integral video support

LUNA

Graphics performance

LUNA is designed to run consistently at 50/60 frames per second, even on low-end devices.

Lightning fast, asynchronous, minimal changes rendering model.

Significantly outperforms HTML5 applications.

LUNA

Application support

Widgets encapsulate common user interface features to facilitate rapid development.

Behaviours automatically support default interface logic for different modes of interaction.

Themes allow rapid changes to visual appearance such as fonts and colours.

LUNA

Flexible and extensible

Ideal for developing television and video user interfaces, including EPG, VOD, or PVR applications.

Not limited to a fixed framework, format or methodology.

Developers can create their own object-oriented modules depending on requirements.

LUNA

Layered approach

Combines declarative and programmatic approaches.

Flexible layered model allows any conceivable user interface to be implemented.

Where a feature is not available in one layer it can be implemented at a lower level.

LUNA Software layers

Declarative
(JSON, JavaScript Object)

- Layout rules
- Theme files

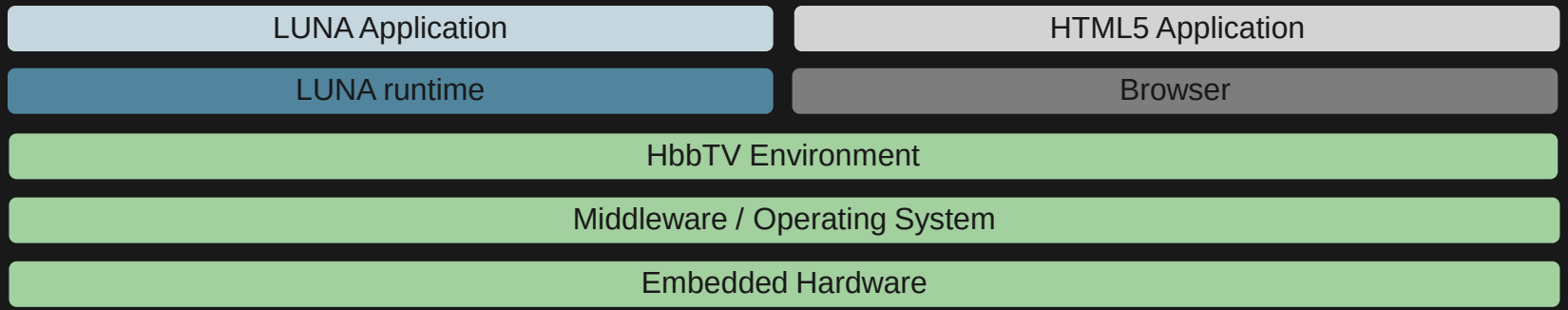
Programmatic
(JavaScript)

- Widgets
- Presentation logic
- Navigation
- Routing
- Data binding

LUNA API

- Text
- Images
- Shapes
- Video

HbbTV integration



LUNA

Layout rules

Responsive layout according to screen resolution, aspect ratio and orientation.

The relative hierarchical position of elements can be absolute or based on rules that automatically adjust to screen sizes and changes in screen orientation.

LUNA

Theme files

Themes can be used to specify the visual presentation, including fonts and colours, allowing the look and feel to be changed easily without affecting the underlying layout or behaviour.

LUNA

Widgets

Library of user interface widgets written in TypeScript, a superset of JavaScript supporting modern software features.

Designed to provide rapid and robust implementation of most essential user interface features for presentation and navigation.

Full source code and examples available.

LUNA

Flexible input

Works with both DPAD cursor button remote control and multi-touch screen interfaces.

- Heuristics provide natural default navigation using Up/Down/Left/Right/Select buttons for any interface.
- Tap and swipe gesture navigation is intrinsically supported for touch screen devices.

LUNA

Data binding

The widgets promote separation of the user interface from the data model using data binding to collections based on JavaScript arrays.

Designed to be used with RESTful backend systems, widgets can be populated by associating them with responses to web requests.

LUNA

Widget library

Containers that can be used to create layouts:

- Canvas
- Flex
- List
- Deck
- Text
- Image
- Video

LUNA

Behaviors

Behaviors can be associated with widgets:

- DPADNavigation
- Multitouch
- Collection
- ScrollBar
- AutoScroll

LUNA

Versatile

The set of widgets and behaviours can be combined to create an almost unlimited range of layouts that address the essential requirements of video user interfaces.

LUNA

Extensible

Many more widgets are in development to address most frequently requested requirements.

Developers can create their own libraries of widgets to enable re-use of components.

LUNA

Development

Simulator available for Windows, Mac and Linux.

Debugger / profiler / development tools available from Google and open-source community.

LUNA

Integration

POSIX, OpenGL ES2 and video player.

Middleware agnostic.

- Integration via REST or DBUS
- C/C++ or Java API bindings to JavaScript

LUNA

Integration

Usually ported to a new architecture in a day.

Already runs on:

- Windows/Mac/Linux
- Android/Android TV
- Apple iOS
- Cisco Fusion
- WyPlay Frog
- Humax Octo
- Raspberry Pi

LUNA

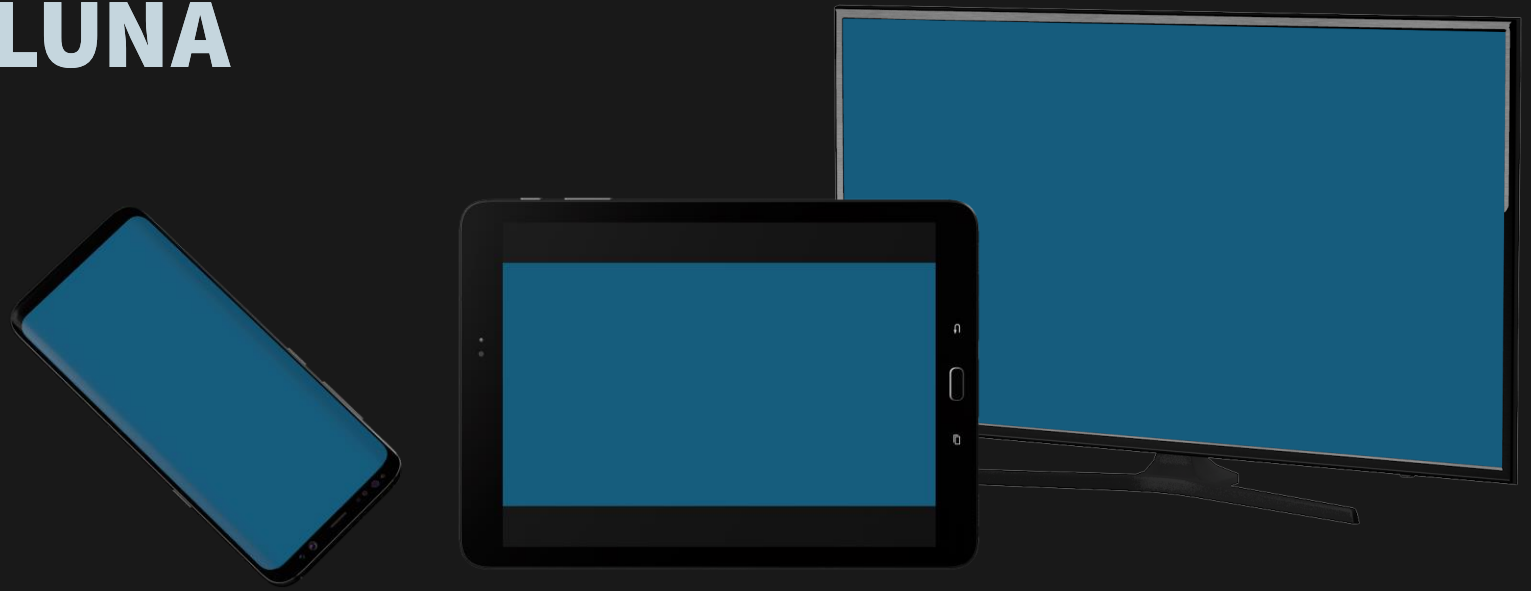
Summary

LUNA provides a powerful cross-platform JavaScript API to high performance graphics rendering in OpenGL.

LUNA can complement HbbTV, providing a highly optimized user experience engine.

Ideal for applications where performance is paramount, such as guides and navigational interfaces.

LUNA



LUNA™ enables *fast, fluid and flexible* visual experiences on any screen

LUNA

Søren Mou Jakobsen
smj@craftwork.dk