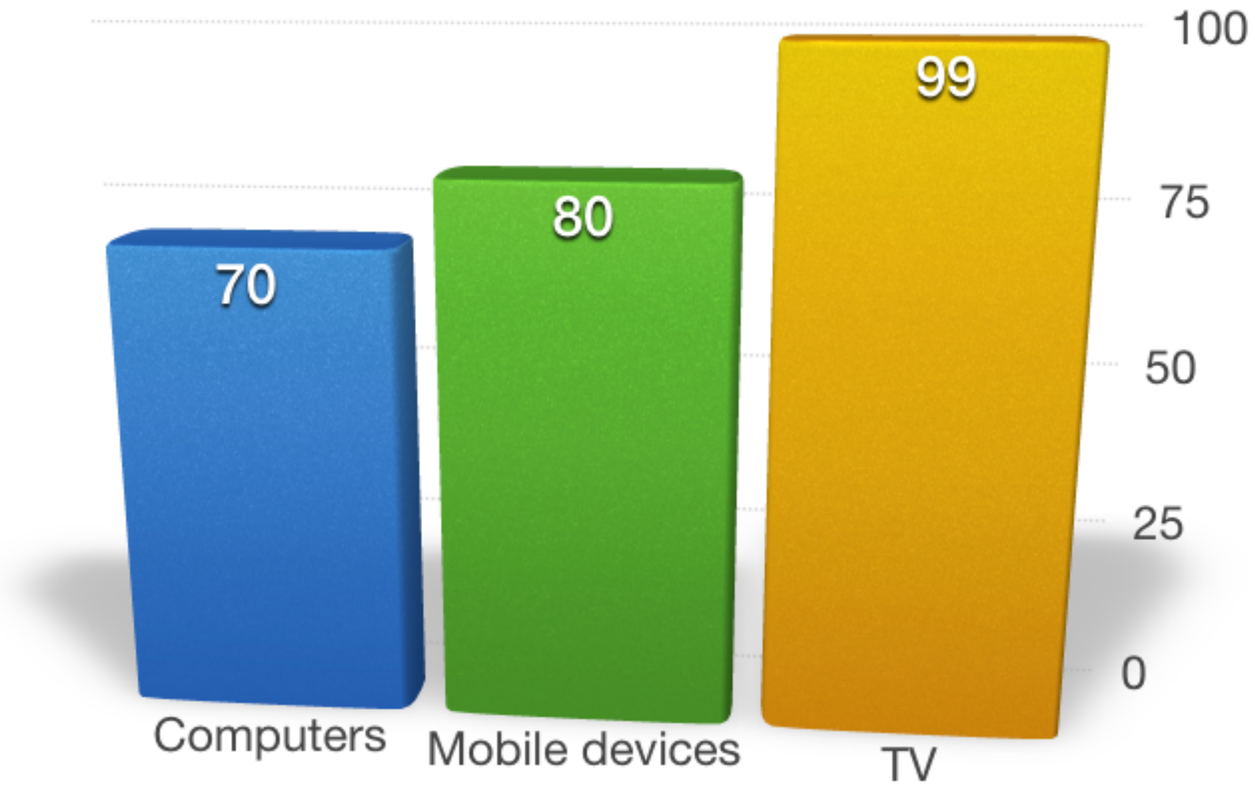
Android TV Game Development

Alexey Rybakov
Senior Developer @ DataArt

Why is TV interesting?

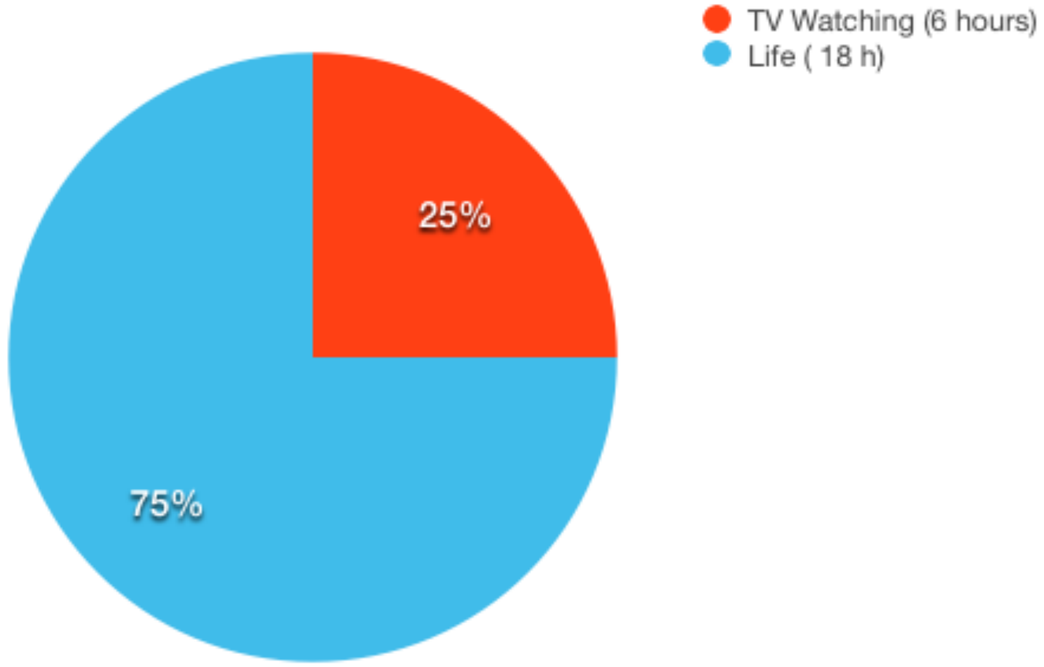
Why is TV interesting?

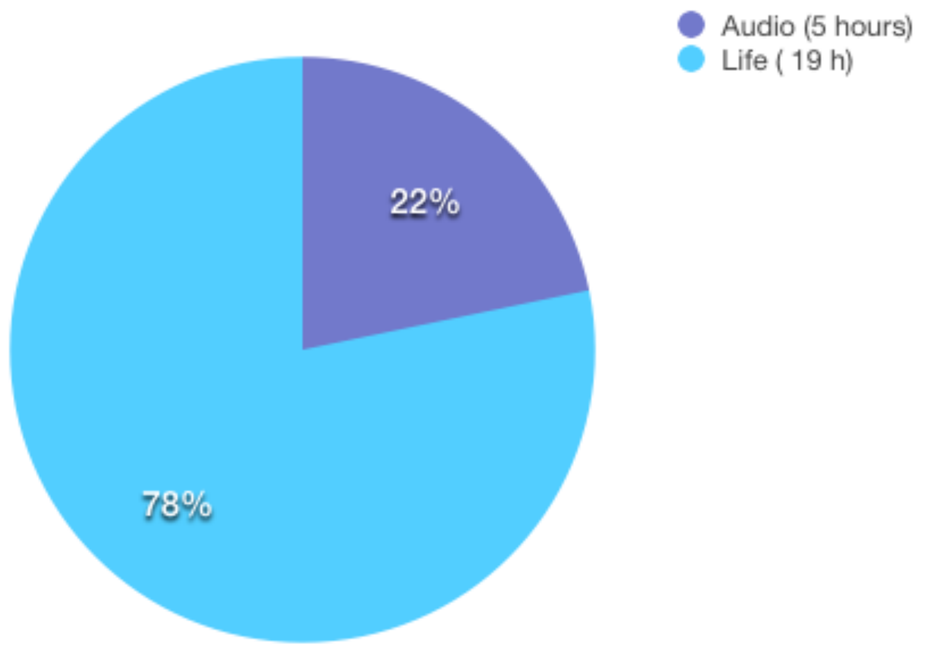
More American households have TVs than cell phones or computers:



Why is TV interesting?
Biggest screen - biggest time

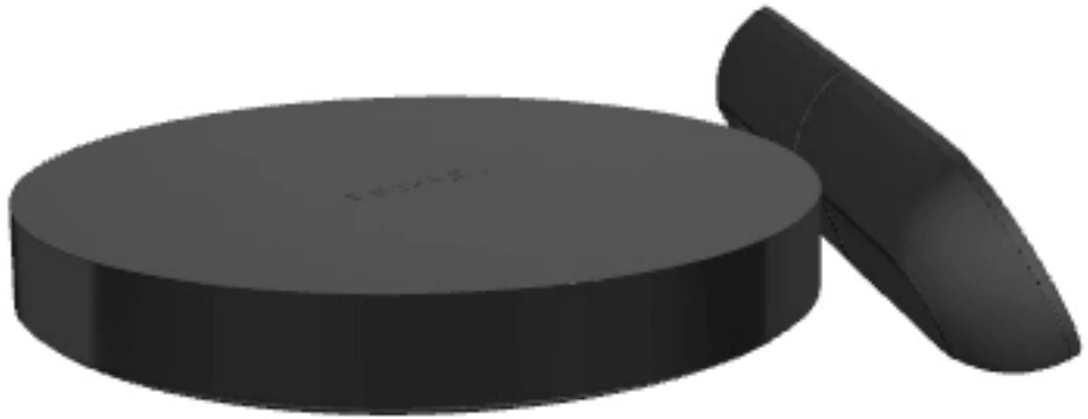
Why is TV interesting?





Android TV Hardware

Nexus Player



CPU: 1.8GHz Quad Core, Intel® Atom™

Memory: 1GB

Storage: 16GB/64GB

Resolution: FullHD

Razer Forage TV

CPU: Qualcomm® Snapdragon™ 805

GPU: Quad-Core Krait 450 CPU

2.5GHz per core

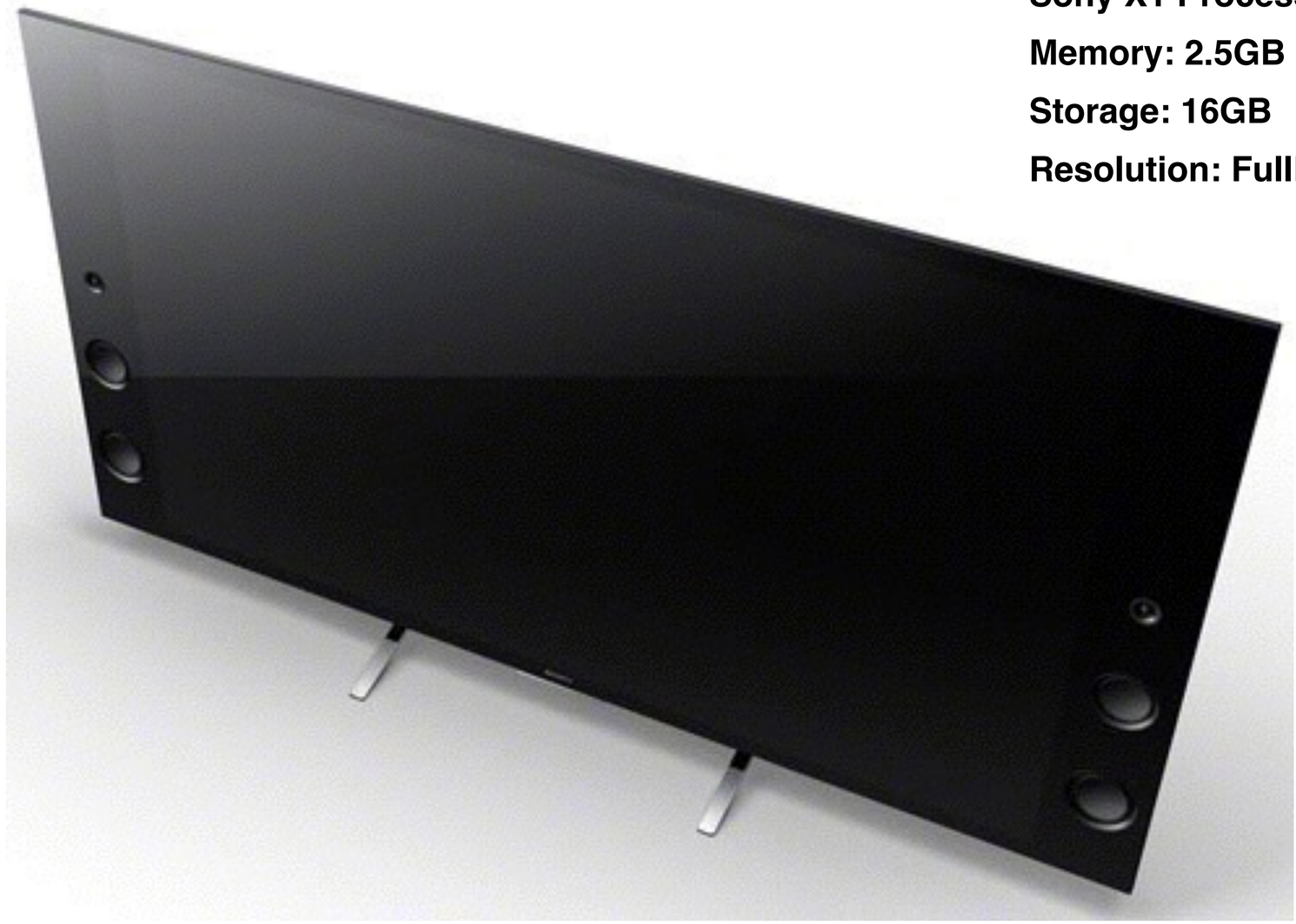
Adreno™ 420 GPU

Memory: 2GB RAM

Storage: 16GB of Storage

Resolution: FullHD

Sony Bravia



CPU: ARMv7 Dual-core CA17

Sony X1 Processor

Memory: 2.5GB

Storage: 16GB

Resolution: FullHD (Scaled to 4K)

Xaomi MI Box

Processor: Quad-core Cortex-A53 2.0GHz

GPU: Mali 450 750MHz

RAM: 2GB DDR3

Flash: 8GB eMMC

Output Resolution: Up to 4K 60fps

NVIDIA Shield

CPU:1 .9 GHz ARM Cortex-A574

GPU: 1000 MHz Maxwell

Memory: 3GB

Storage: 16GB/500GB

Resolution: 4K

Tegra X1

TEGRA X1 PROCESSOR SPECIFICATIONS	
	TEGRA X1
GPU	**NVIDIA Maxwell 256-core GPU** DX-12, OpenGL 4.5, NVIDIA CUDA®, OpenGL ES 3.1, AEP, and Vulkan
CPU	**8 CPU-core, 64-bit ARM® CPU** 4x A57 2MB L2; 4x A53 512KB L2
VIDEO	**H.265, VP9 4K 60 fps Video** 4k H.265, 4k VP9, 4k H.264
POWER	**20 nm SOC - TSMC** Isolated Power Rails, Fourth-Generation Cluster Switching
DISPLAY	**4K x 2K @60 Hz, 1080p @120 Hz** HDMI 2.0 60 fps, HDCP 2.2

Android SDK Game Features

4K Support

- **Display.getSupportedModes()**
  - Returns the available modes, including 4K is supported
  - Includes IDs representing the modes
- **WindowsManager.LayoutParams.prefferedDisplayModeId**
  - Request a desired mode
  - Affects **SurfaceViews** in the window
  - But SurfaceViews are used for 3D content
- This API does not always return 4K support on devices that can

4K Native Activity Support

- 4K may still be available
  - If the user has selected 4K in the device's settings
  - **com.android.tv.settings/ .device.display.hdmi.HdmiActivity**
- Test for the availability of 4K via the property
  - **sys.display-size**
- Set the back buffer to 4K via the NDK call
  - **ANativeWindow_setBuggersGeometry**

HDMI Settings

Adoptable Storage

- Was added to make SD cards more useful in Android
- Allows more type of data onto expandable storage
  - But also encrypts cards
- App storage paths change dynamically
- This broke a lot of applications
- App must use the Context and ApplicationInfo to get data paths
  - And should not cache the paths

Low Latency Audio

- Audio latency has been improved
- **https://github.com/googlesamples/android-audio-high-performance/**
- Tag: **android.hardware.audio.low_latency**
  - professional audio tools
  - Pro-audio developers

Gamepads

- Not all gamepads the same
- Test for available axes/buttons
- Show controllers tips
- Detect hot plug/unplug

The Middle Buttons

Left Most	see notes *or* Button11
Middle	KEYCODE_HOME
Right Most	*Button10*

Button placement is typical but not absolute.

Left Most button doesn't generate a typical event. To check if back was pressed, use this code:

```
if (Input.GetKeyDown(KeyCode.Escape))
   { // BACK pressed };
```

This button can also generate Button11, it will never generate both (Button11 and the above code).

Button should be used as "go back" or "pause/options" in your application. If you capture the event, you should make sure it makes sense in your application given the current context, see "In the UI . . ." above.

Middle button should be ignored. It is used for OS functionality.

Right Most, while on most controllers, is *optional*, button should be used for in-application "pause/options" menu when present.

See "DPAD and System Buttons" below.

The DPAD

U	Axis6	-1.0
D	Axis6	1.0
L	Axis5	-1.0
R	Axis5	1.0

See "DPAD and System Buttons" below.

The Bumpers and Triggers

Left Bumper	Button4
Right Bumper	Button5

Depending on the physical controller, there are 3 possible input values for triggers, see below.

Left Trigger

1	Axis13	0.0 to 1.0
2	Axis7	0.0 to 1.0
3	No TRIGGER on Controller	see notes

Right Trigger

1	Axis12	0.0 to 1.0
2	Axis8	0.0 to 1.0
3	No TRIGGER on Controller	see notes

Notice that some controllers don't have triggers, Scenario 3 above. While controllers without triggers are a small subset, consider allowing those functions on other buttons.

However, if your application *requires* triggers to be available for a first class user experience, you should consider ignoring those controllers.

In some cases, a controller will seem to issue two trigger events for a single press – Axis12/Axis8 or Axis13/Axis7. Both are identical in value. You need to handle the one with a value.

To give you an idea on one way to handle the above events, consider this pseudo-code:

```
OnMotionEvent
   RTvalue = MAX(Axis12, Axis8);
   LTvalue = MAX(Axis13, Axis7);
```

This will require the Input Manager to be setup in a particular way. See the "Unity API Specifics" section below.

The Analog Sticks

LS: L2R	AxisX	-1.0 to 1.0
LS: U2D	AxisY	-1.0 to 1.0
LS: Pushed	Button8	n/a
RS: L2R	Axis3	-1.0 to 1.0
RS: U2D	Axis4	-1.0 to 1.0
RS: Pushed	Button9	n/a

The Action Buttons

Y button	Button3
B button	Button1
A button	Button0
X button	Button2

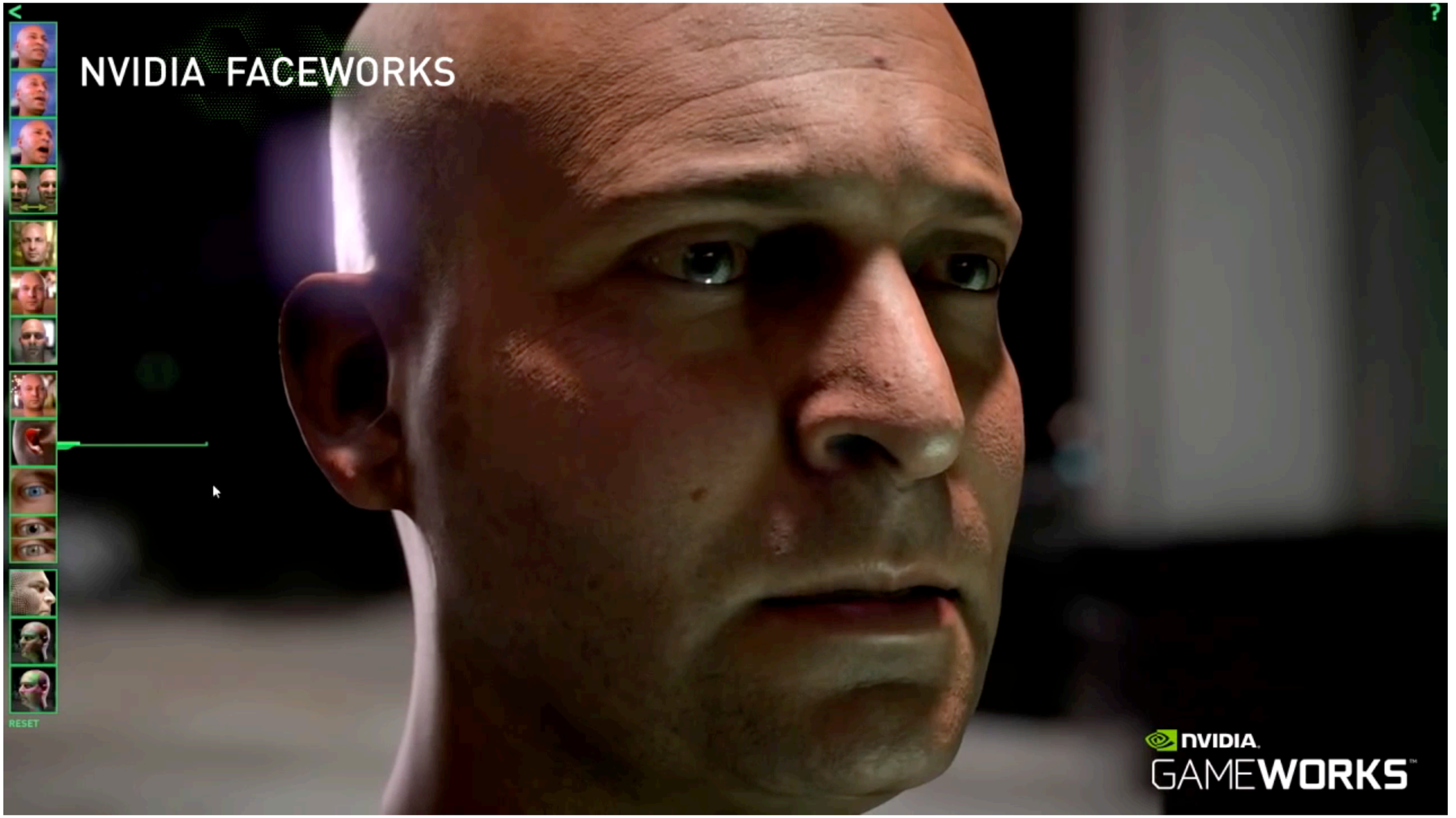
Ignore the printing on your testing gamepad. The above standard should be your reference.

Gamepads:

virtual bool	**pollGamepads** (AInputEvent *event, uint32_t &changedMask) Passes Android input events to the gamepad system for processing.
virtual bool	**getState** (int32_t padID, **State** &state) Get the state of a gamepad.
virtual void	**setMaxGamepadCount** (int32_t max) Sets the maximum number of simultaneous gamepads.
virtual int32_t	**getMaxGamepadCount** () Gets the maximum number of simultaneous gamepads.

uint32_t	**mButtons** Flag vector of BUTTON_*, with a bit set to 1 indicating the corresponding button is pressed.
float	**mLeftTrigger** Left analog trigger value [0.0, 1.0].
float	**mRightTrigger** Right analog trigger value [0.0, 1.0].
float	**mThumbLX** Left thumbstick horizontal axis [-1.0, 1.0], deadzone already zeroed.
float	**mThumbLY** Left thumbstick vertical axis [-1.0, 1.0], deadzone already zeroed.
float	**mThumbRX** Right thumbstick horizontal axis [-1.0, 1.0], deadzone already zeroed.
float	**mThumbRY** Right thumbstick vertical axis [-1.0, 1.0], deadzone already zeroed.

NVIDIA GameWorks


NVIDIA FACEWORKS
RESET
NVIDIA
GAMEWORKS

NVIDIA FaceWorks

- High Quality Skin Shaders (skin subsurface scattering)
- Deep Scattering (Light transmission through thin membranes)
- Eye refraction (upcoming)

- Platforms PC
- Dependencies DX11

NVIDIA Wave Works

- Tessendorf's spectral algorithm, based on Phillips spectrum
- Multi-res simulation
- Quad-tree tile-based LoDing
- Host read-back
- DX11 tessellation
- Foam simulation
- A "no graphics" path for clients (MMO servers)

Platforms PC, Steam OS, Linux, MacOS, PS4, XBOX1
Dependencies DX11
Engines UE4 (GitHub)

NVIDIA Flex

- Artist-focused tools to ensure turnkey solutions
- Unified solver for effects
- Rigid/deformable bodies
- Phase transition
- Particles
- Fluids
- Cloth
- Rope
- Adhesion
- Gases

Platforms Win/Linux
Dependencies TBA
Engines UE4 (GitHub)

NVIDIA Volumetric Lighting

- Physically based light scattering
- Adds atmospheric depth
- Dramatic light shafts
- Highly tunable

- Platforms PC, Console (inquire)
- Dependencies DX11
- Github Source on Github

NVIDIA Turbulence

- Artist-focused tools to ensure turnkey solutions
- Interactive massive particle simulation
- Heat sources and jets
- Integrated with PhysX Particles
- Noise support
- Ease of scalability for different gaming platforms
- Level of Detail control

- Platforms PC
- Dependencies APEX framework
- Engines UE3, UE4

NVIDIA Hair Works

- Key Features
- Supports off-the shelf grooming tools
- Shape & style control
- Self shadowing
- Body to hair shadow casting
- Wind interaction
- Level of Detail
- Scalability
- Real time editing in viewer

- Platforms PC
- Dependencies DX11
- Tools 3dsMax/Maya/Previewer

NVIDIA PhysX SDK

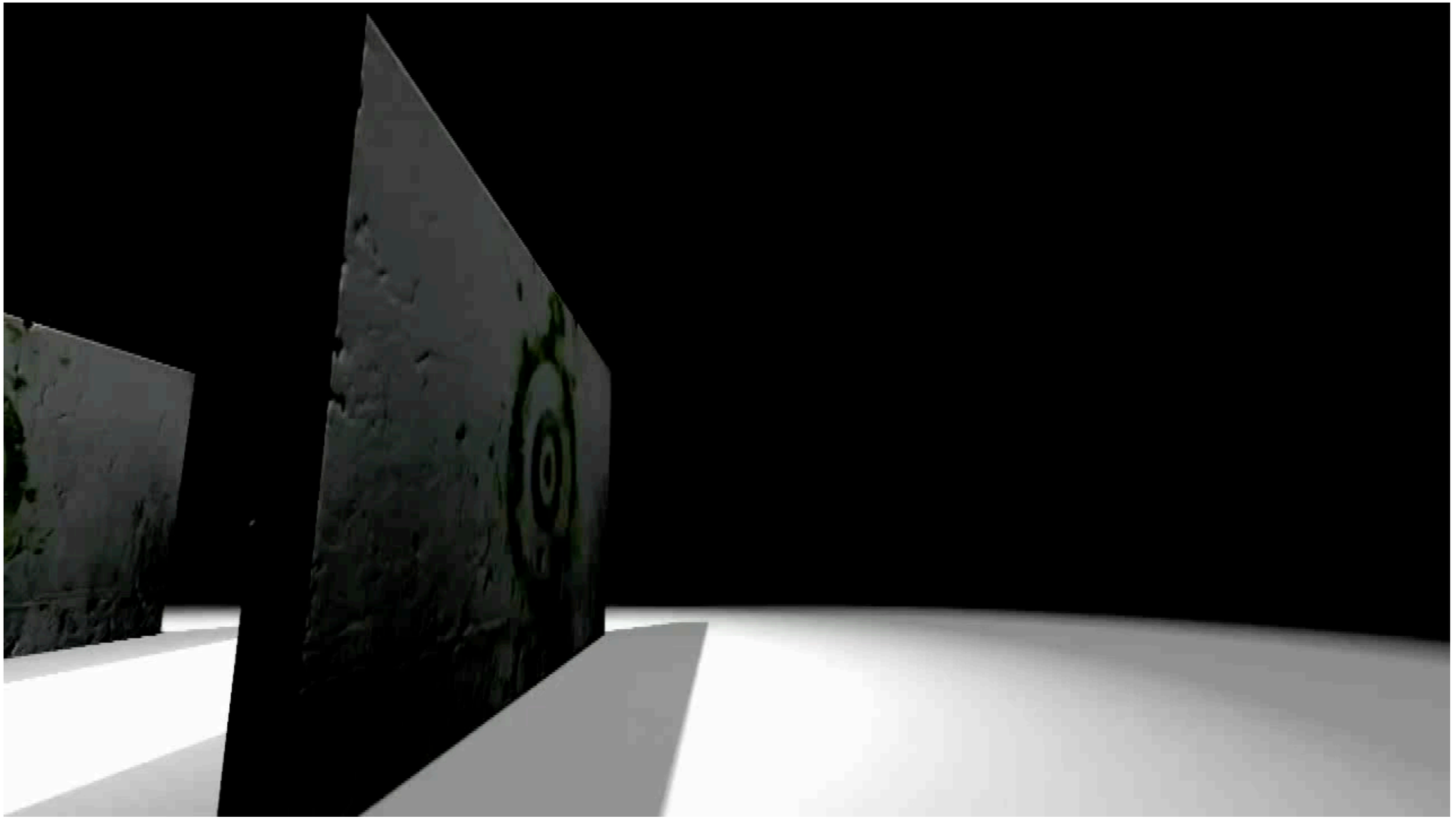
- Rigid Body Simulation
- Collision Detection
- Character Controller
- Particles
- Vehicles
- Cloth

- Platforms Win, OSX, Linux, XBOX®, PlayStation®, Android, ioS
- Engines Unreal® Engine 3, Unreal® Engine 4, Unity®
- Tools PhysX Visual Debugger
- Maya DCC Plug-In
- 3D Studio MAX DCC Plug-In

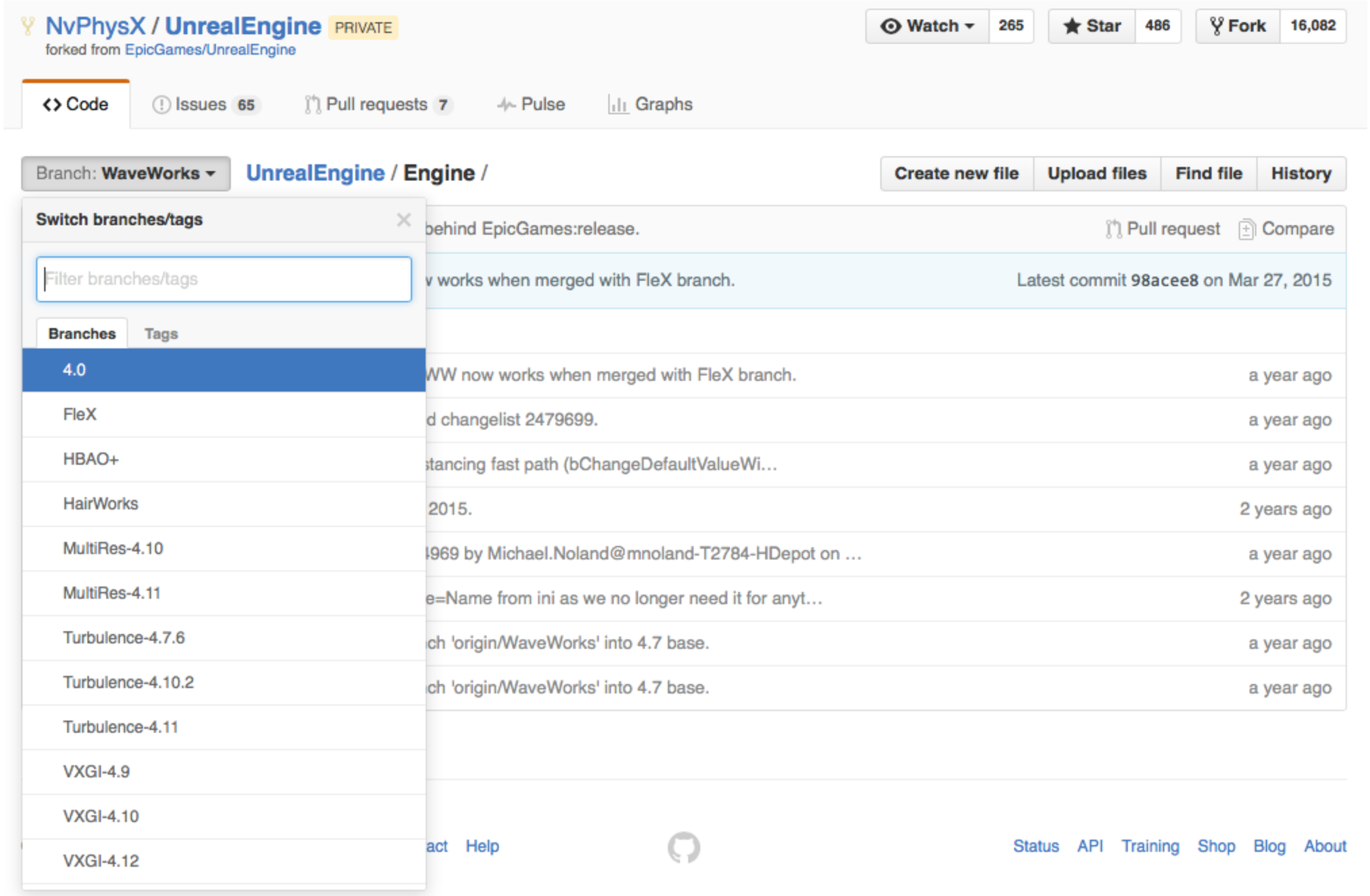
NVIDIA PhysX SDK

- Artist-focused tool to ensure turnkey solutions
- Full and partial destruction support (gameplay or non gameplay-affecting)
- Support for GPU-accelerated Rigid Body debris
- Ease of scalability for all supported gaming platforms
- Network Support
- Level of Detail - the amount of persistent destructible debris can be scaled up or down

- Platforms PC, XBOX, PS, Android
- Dependencies PhysX, APEX framework



GameWorks: open source



GameWorks: Links

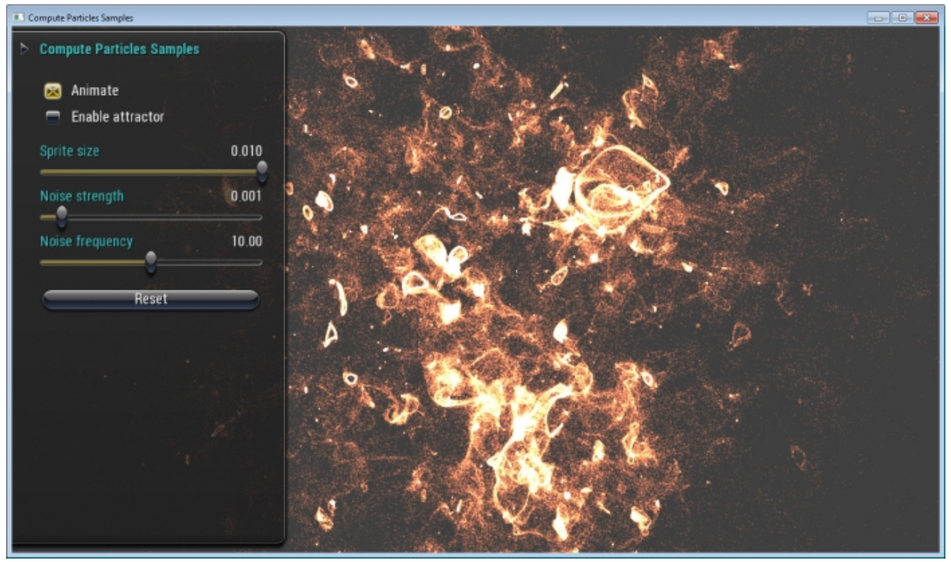
https://github.com/NVIDIAGameWorks

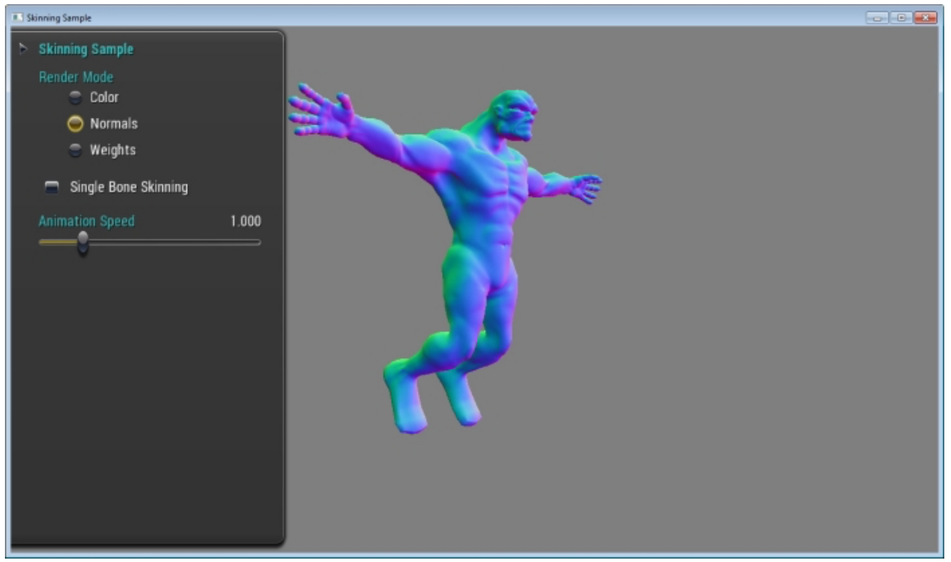
https://github.com/NvPhysX/UnrealEngine

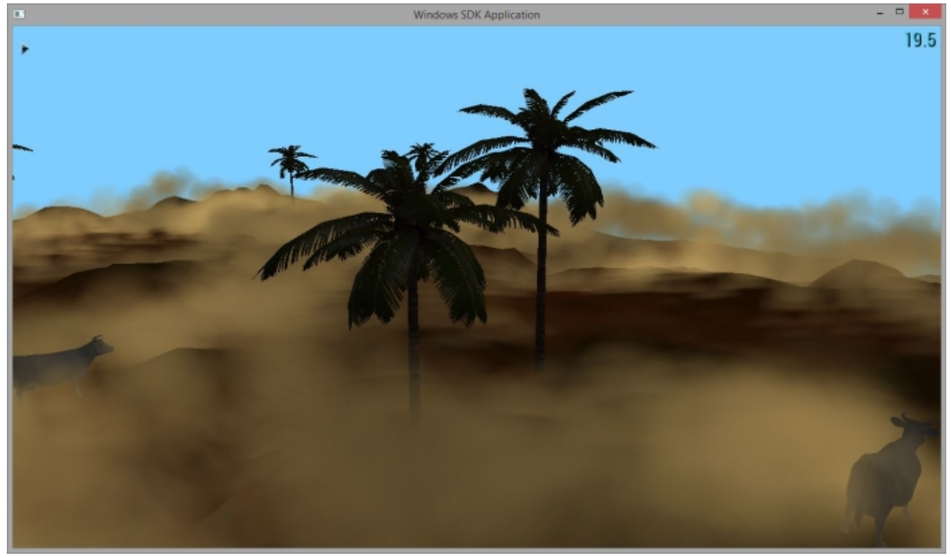
https://github.com/nvpro-samples

https://github.com/NVIDIA

Samples


Compute Particles Samples
Compute Particles Samples
Animate
Enable attractor
Sprite size
0.010
Noise strength
0.001
Noise frequency
10.00
Reset


Skinning Sample
Skinning Sample
Render Mode
Color
Normals
Weights
Single Bone Skinning
Animation Speed
1.000


Windows SDK Application
19.5

Vulkan

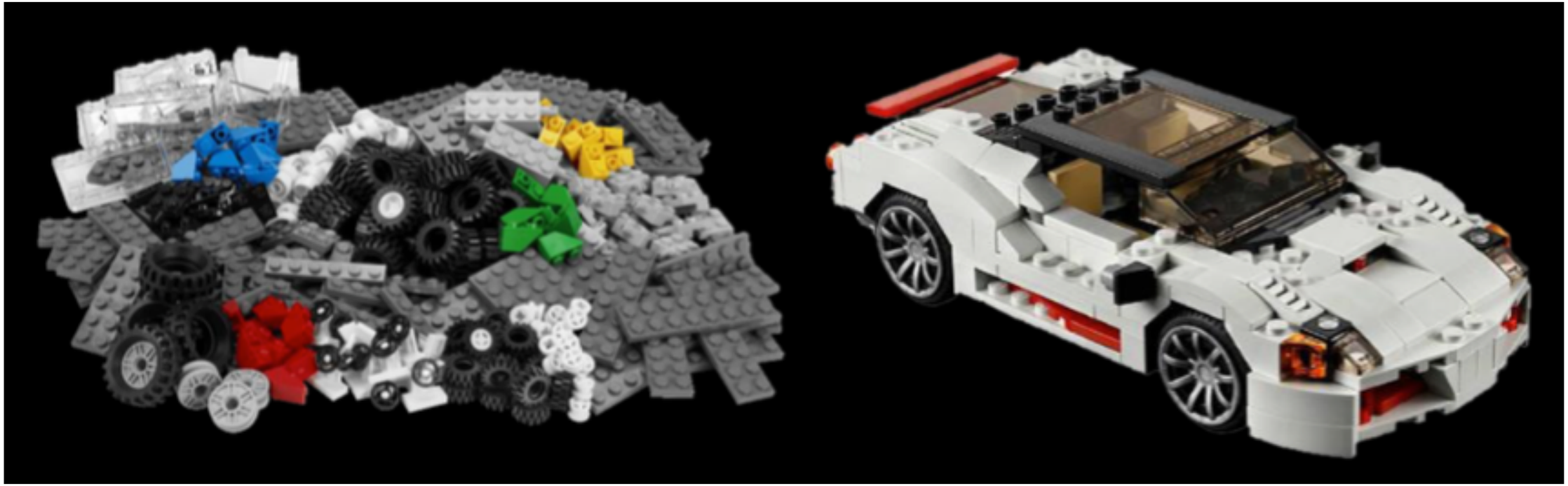
Vulkan

- New open standard for high-performance 3D rendering
- Threaded friendly
- Efficient

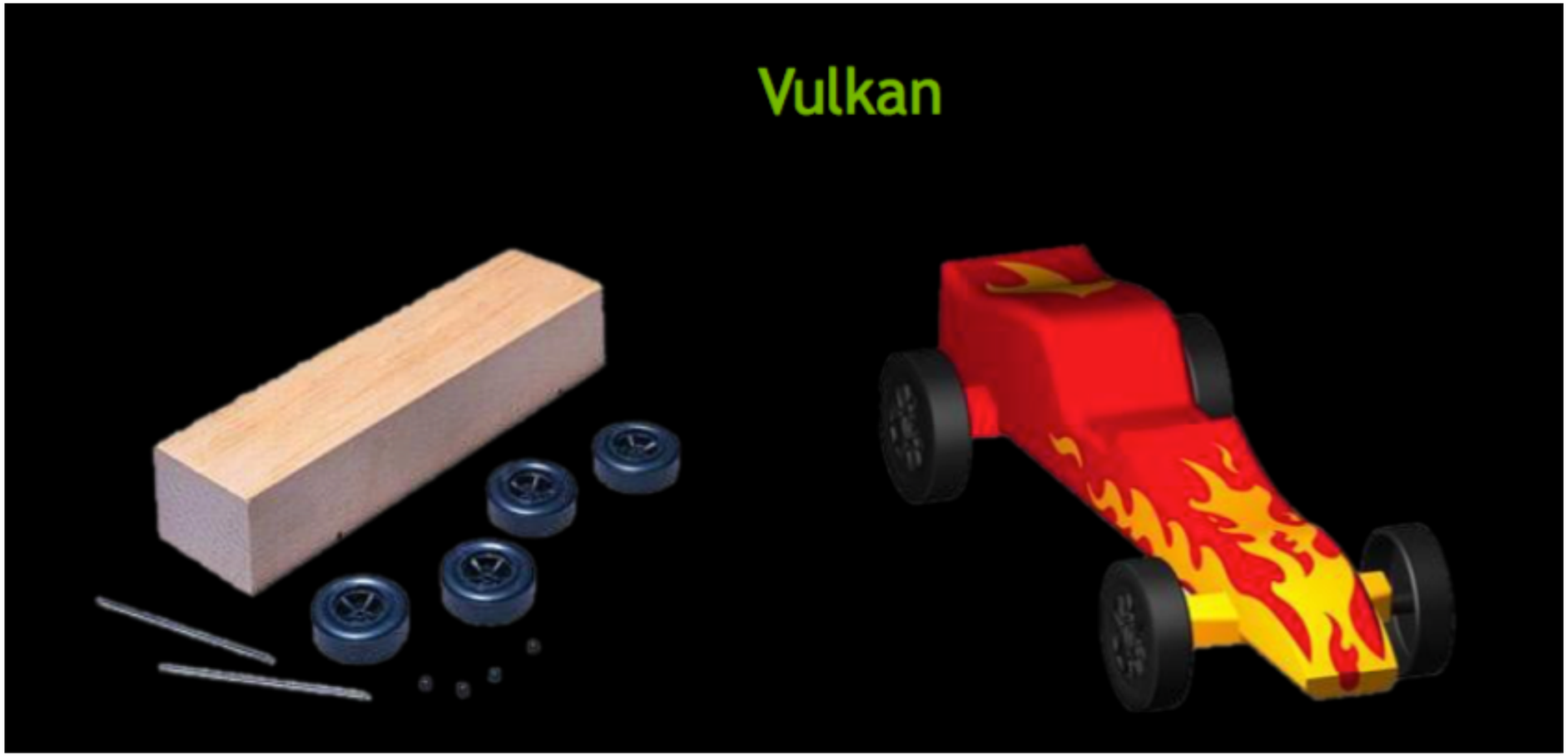
Analogy

Fixed-function OpenGL

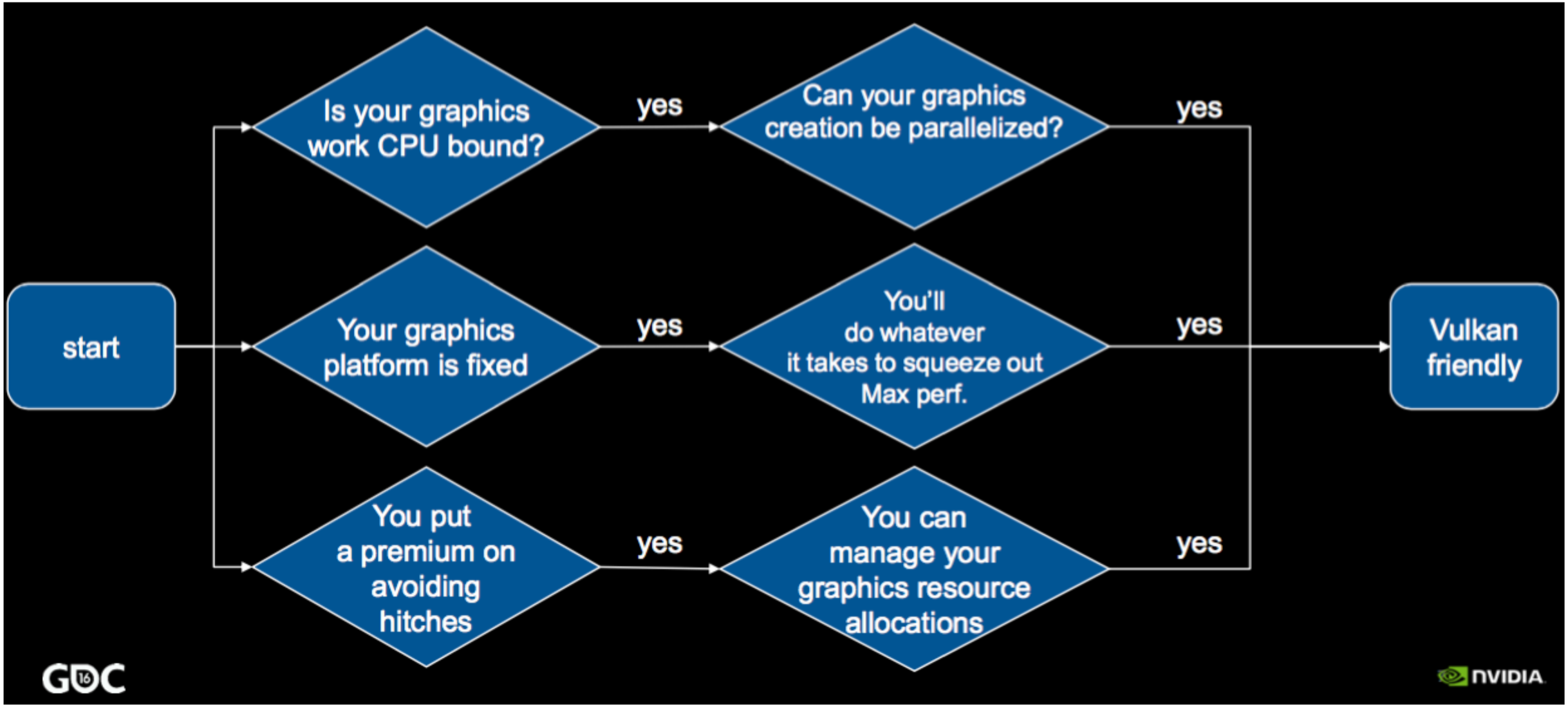
Modern AZDO OpenGL with Shaders



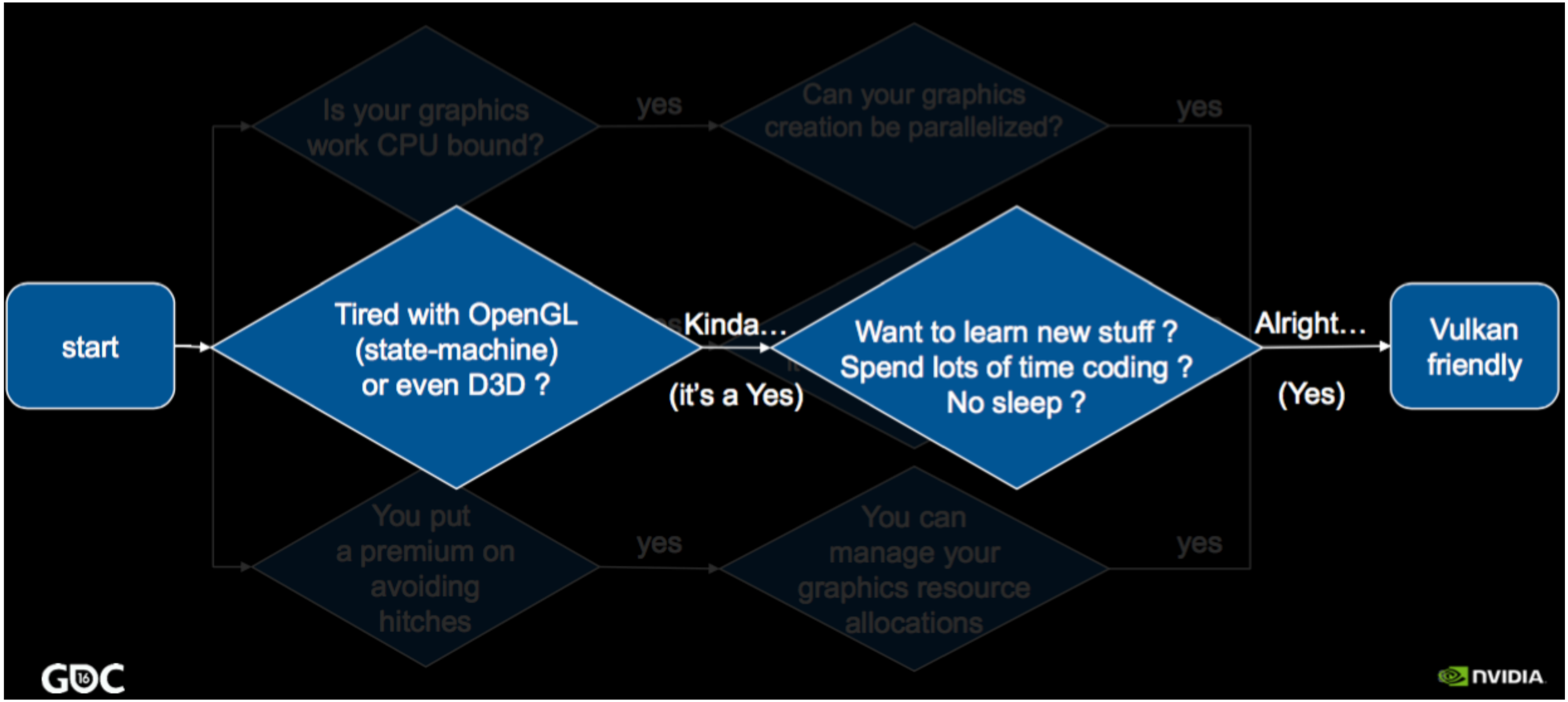
Vulkan



You need Vulcan if



You need Vulcan if



More Info:

https://developer.nvidia.com/gdc-2016

Vulkan and NVIDIA – The Essentials

In this talk we'll discuss Vulkan and NVIDIA.

- Tristan Lorach

File: Vulkan and NVIDIA – The Essentials
Video

High-Performance, Low-Overhead Rendering with OpenGL and Vulkan

Welcome to our talk about High-performance, Low-Overhead Rendering with OpenGL and Vulkan. Lars and I are with NVIDIAs developer technology teams. He's been focusing on mobile side things and I've been focusing on the desktop side of things.

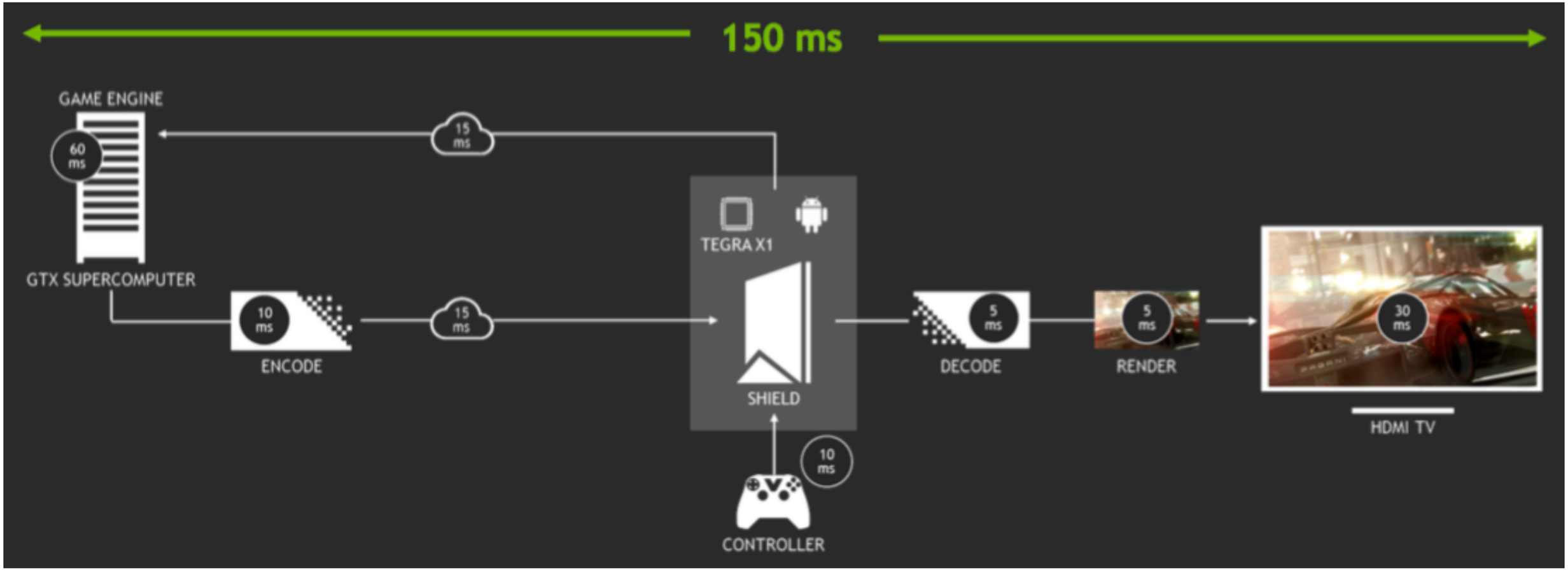
- Mathias Schott (Sr. Developer Technology Engineer, NVIDIA)
- Lars M. Bishop (Sr. Developer Technology Engineer, NVIDIA)

File: OpenGL and Vulkan pdf
Video

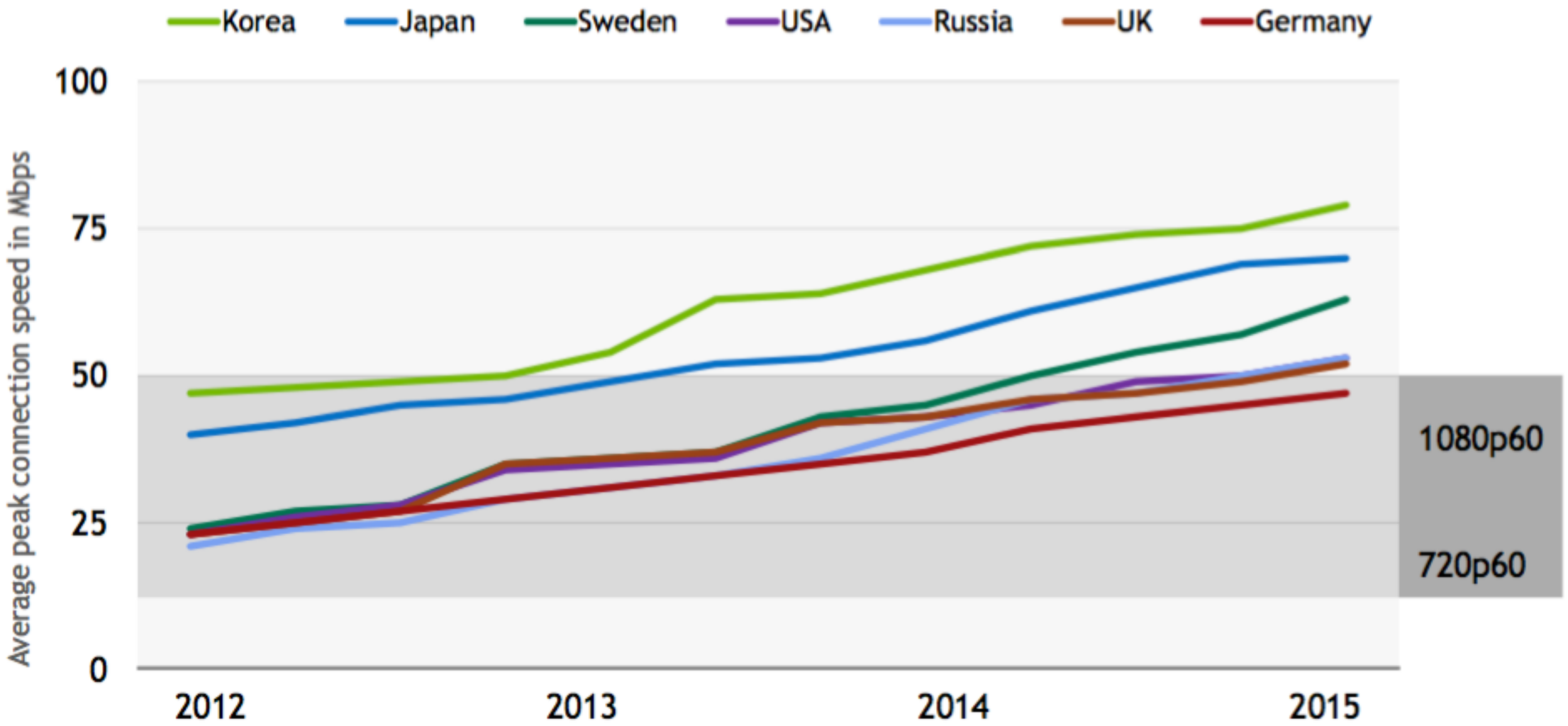
GeForce Now


TOMB RAIDER
PLAY IT ON SHIELD™.
SQUARE ENIX
CRYSTAL DYNAMICS
NVIDIA. SHIELD
An NVIDIA® GeForce Now™ membership is required to play or purchase games.

How GFN Works



Global Broadband: Doubles Every 3 Years



Source: Akamai's State of the Internet Reports

GeForce Now Performance

Performance relative to Xbox One

Requirements

	REQUIRED	DESIRED
Operating System	Windows 7+	Windows 7+
API	DirectX 9/10/11	DirectX with GameWorks
Controller	Gamepad	And keyboard/mouse
Resolution	720p	1080p
Refresh	30fps	60fps
Audio	Stereo	5.1 Surround
Ratings	ESRB	ESRB/PEGI/USK

Add Game to GFN: 4 steps

Register Account

Describe Game

Upload Binaries

Publish Game

More GFN Information

Email: GFNLink@nvidia.com

Access to GFN link: https://github.com/NVIDIA/GFN-Link

Thank you!

Questions?

Alexey Rybakov
DataArt