

The logo for the GPU Technology Conference, featuring the text "GPU TECHNOLOGY CONFERENCE" in white on a green rectangular background. The background of the entire slide is a vibrant, abstract digital grid with glowing lines in shades of blue, green, yellow, and red, creating a sense of depth and connectivity.

GPU TECHNOLOGY
CONFERENCE

Programming Project Shield and Tegra 4

Andrew Edelsten (Manager, Tegra Developer Technologies)
Richard Seis (Senior Engineer, Tegra Developer Technologies)

Overview

- Tegra 4
- Project SHIELD
- NVIDIA development tools for Android
 - Samples and base NativeActivity projects
 - Nsight Tegra, Tegra Profiler & PerfHUD ES
- Game considerations for Project SHIELD

Tegra 4

“NVIDIA Tegra 4 is a promising processor that’s going to bring a whole *new level of gaming* to mobile devices.”



“If you enjoy the *web browsing* experience on your iPad, you’re going to be pretty pleased what NVIDIA has to offer here.”



Hottest gadgets MWC 2013

Better photography: The CNN logo, consisting of the letters "CNN" in a white, bold, sans-serif font on a red rectangular background, with a white globe icon to the right of the letters.

“If you want to take *better pictures* on your mobile device, NVIDIA’s Chimera computational photography engine is the technology you’ve been waiting for.”

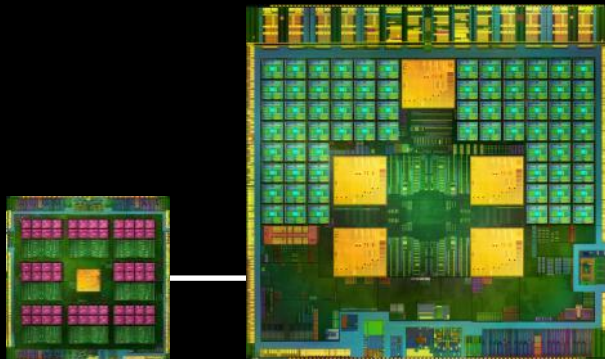


Tegra 4 Family

Tegra 4 (“Wayne”)

World’s Fastest Mobile Processor

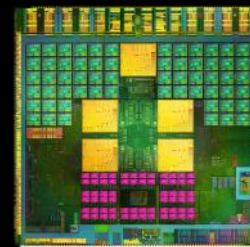
Superphone / Tablet



Tegra 4i (“Grey”)

1st Integrated Tegra 4 LTE Processor

Smartphone



Quad CPU
NVIDIA GPU
LTE
Chimera*

Cortex A15, 4+1

72 Core

Optional with i500



Cortex A9 r4, 4+1

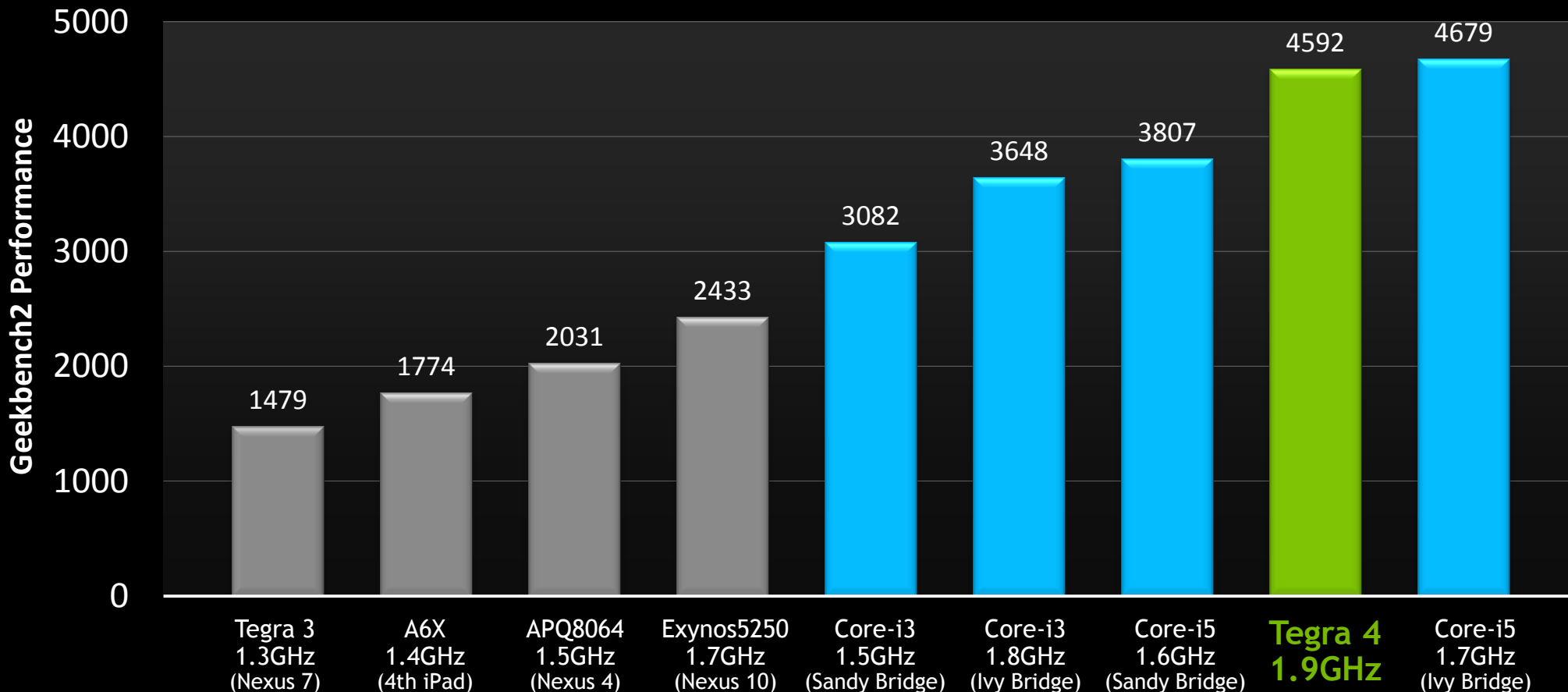
60 Core

Integrated i500



* Chimera is NVIDIA's Computational Photography

Mobile Processor, Ultrabook Performance



Intel Core i3-2377m 1.5GHz, Core i3-3217U 1.8GHz & Core i5-2467m 1.6GHz, Core i5-3317U 1.7GHz all have 17W maximum TDP
 Competitive data published on Geekbench website; Tegra 4 1.9GHz measured on reference platform



nVIDIA®

Project SHIELD

Project SHIELD

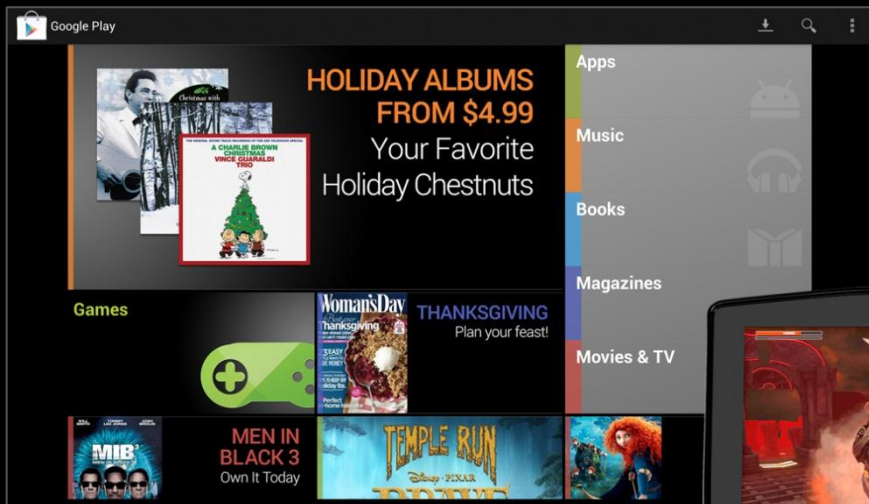
- Tegra 4 powered
- 5 inch 720p & multitouch display
- Console grade controller
- High speed Wi-Fi
- Full connectivity (HDMI, USB, microSD, headphone)
- Pure Android (currently Jellybean)



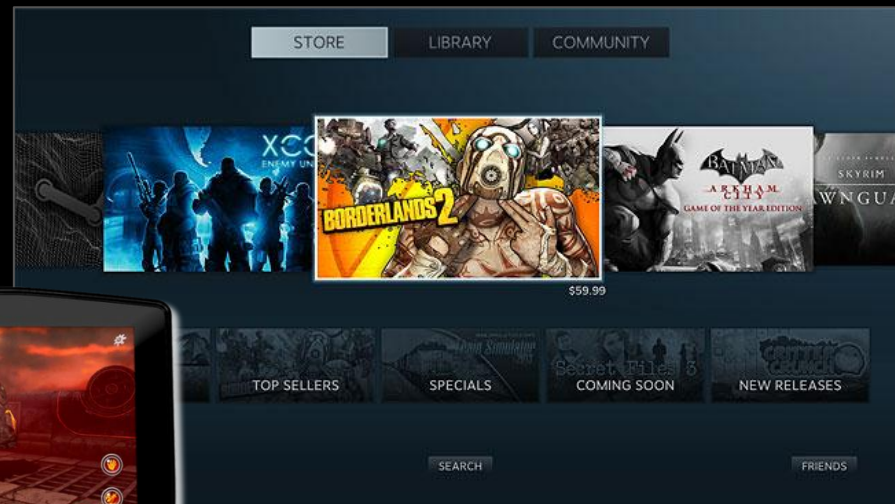


Tuned Port, Bass Reflex Speakers

Two Open Platforms - One Amazing Portable



Android



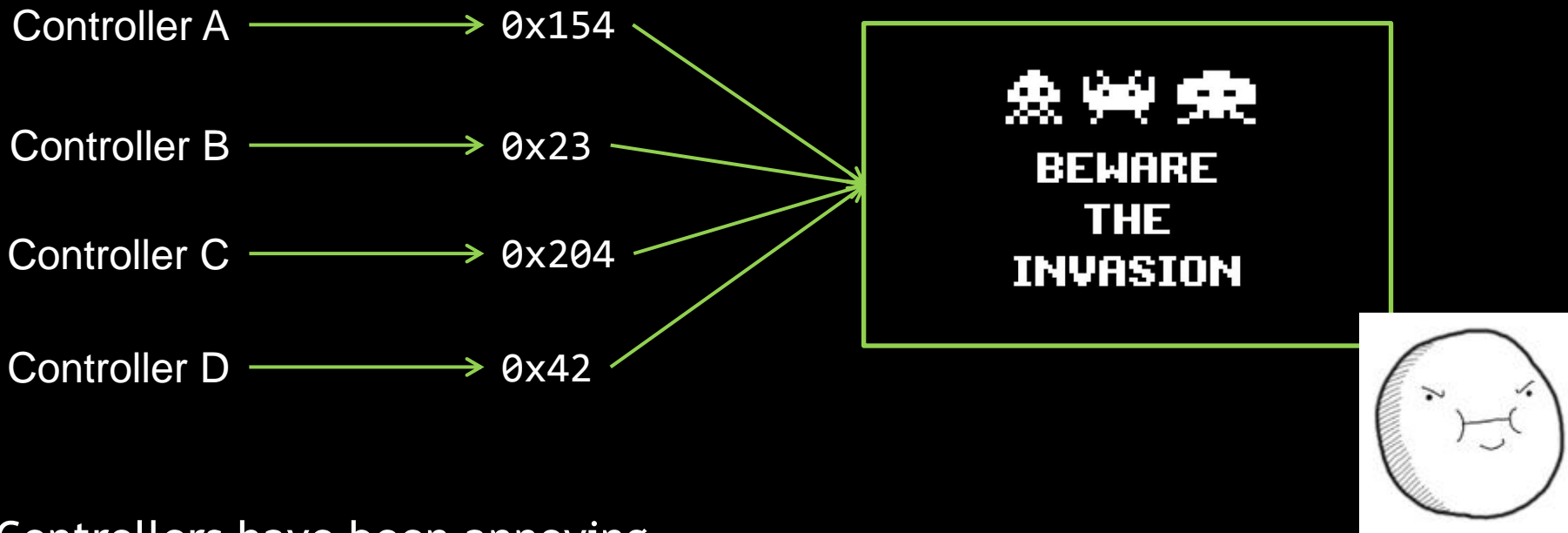
PC



Controlling the Controller

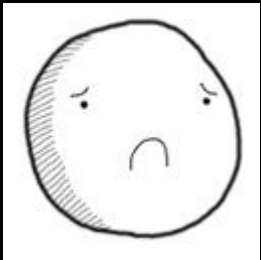
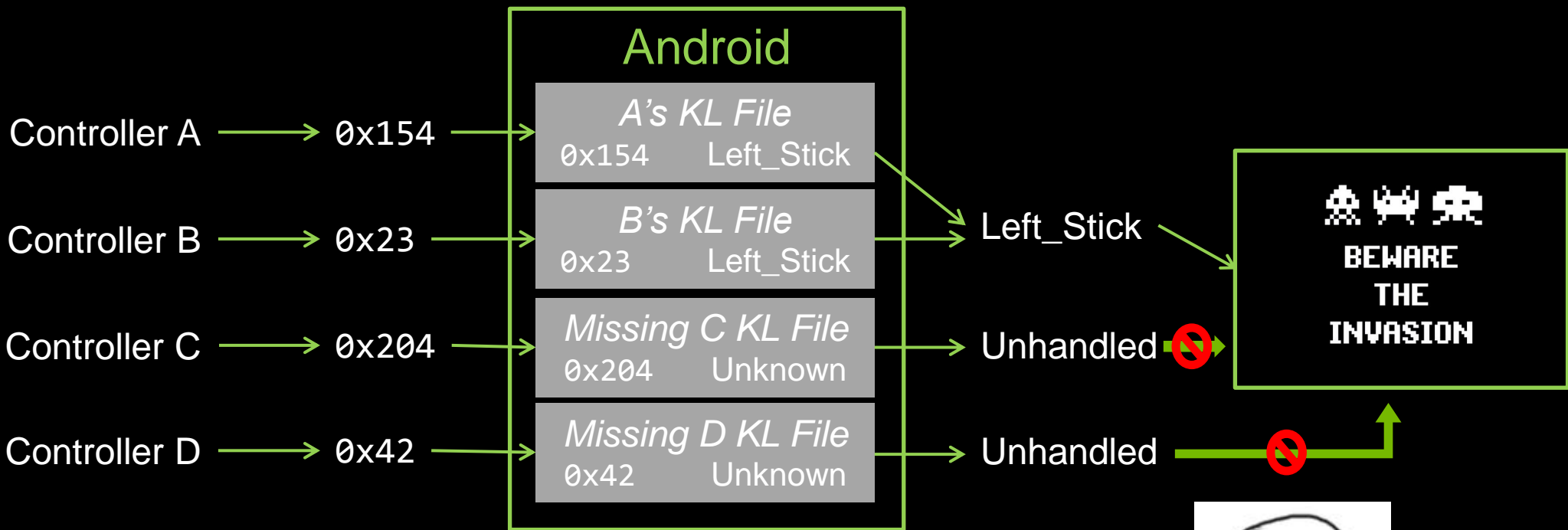
Richard Seis (Senior Engineer, Tegra Developer Technologies)

Controllers: Quick Overview



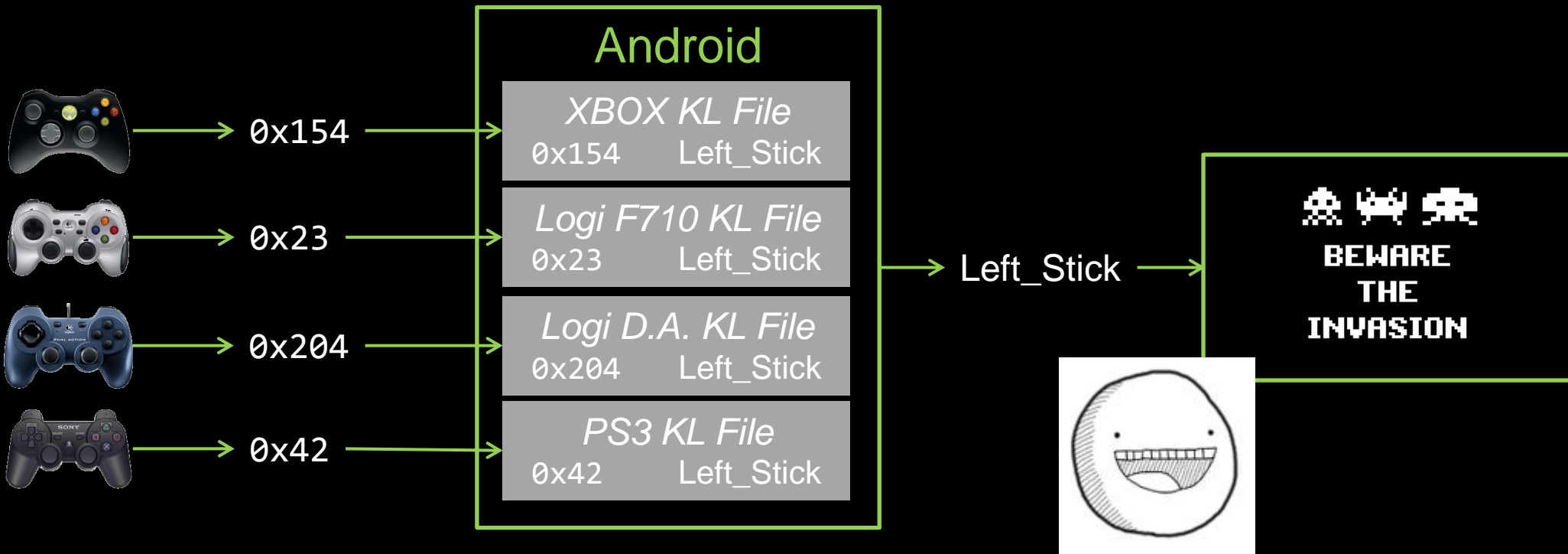
- Controllers have been annoying
 - There is no controller standard
 - Controllers can output what they like
- Supporting multiple controllers like this is a real headache

Controllers: Quick Overview in Android



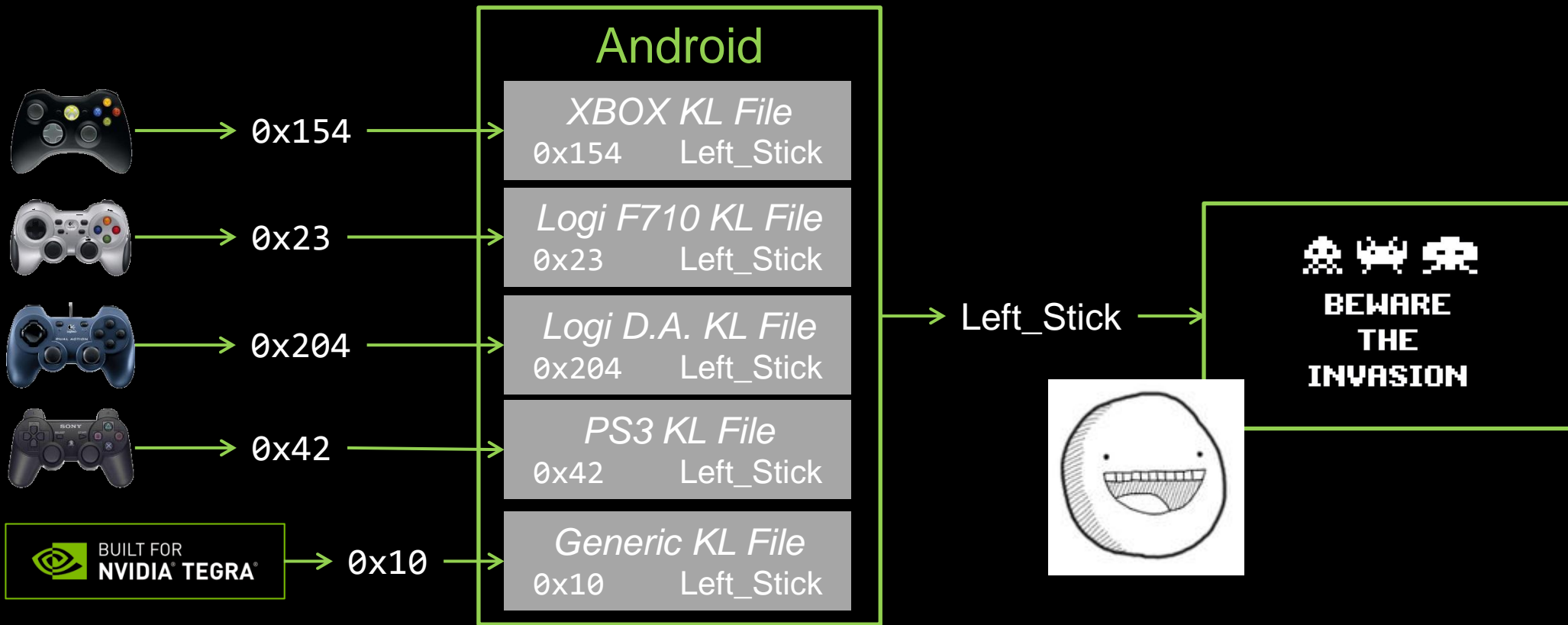
- Android has the ability to help
 - But few translation (“KL”) files
- Supporting multiple controllers like this still a headache

Controllers: NVIDIA is helping



- Since ICS, included KL's for popular controllers
- We have documentation on this normalization
- Supporting multiple controllers like this still a headache

Controllers: Built for NVIDIA Tegra



- Not a proprietary standard
- Developed for controller manufacturers
- Normalization at the hardware level
- See native_gamepad sample for correct input handling - <http://developer.nvidia.com/tadp>

Controllers: Your Game

- Games under 20fps feel sluggish
- Test your game using HDMI too
 - Look at big screen and small screen
 - Performance hit
- Auto-detect the controller and use it
- You can do multiple controllers!
- Explain complex controls
- Remove on-screen controls
- Code to the normalized controller
 - Built for NVIDIA Tegra



Controllers: Your UI

- Don't forget your UI!
- Have a visual indicator of focus
- Use classic standards for navigation
 - 6 and 9 o'clock - Yes
 - 12 and 3 o'clock - No
- Every function must be usable
 - Sliders, buttons, etc.
- Have EXIT item on Main/Pause menus
 - User may be 10' away from touch screen



Questions?

- Andrew Edelsten
- Richard Seis

- NVIDIA Developer Zone
 - <http://developer.nvidia.com/develop4tegra>

- Next up in this room:
 - Paul “Hodge” Hodgson with “Optimizing Tegra Apps and Games Using Unity”
 - Stephen Jones with “Performance & Debugging Tools for Tegra”