You've got the power: your game running better on portable devices

Antoine Cohade
Graphics Application Engineer – Intel

GAME DEVELOPERS CONFERENCE EUROPE COLOGNE, GERMANY · 15-16 AUGUST 2016

Agenda

- Why power matters
- Exploring power efficiency
- How to optimize your game
- Practical case: Lego Minifigures

Why power matters

- User reviews:

"Kills the battery."

*"This app is a **huge battery drain**."*

*"Can **kill your battery** in a blink of an eye."*

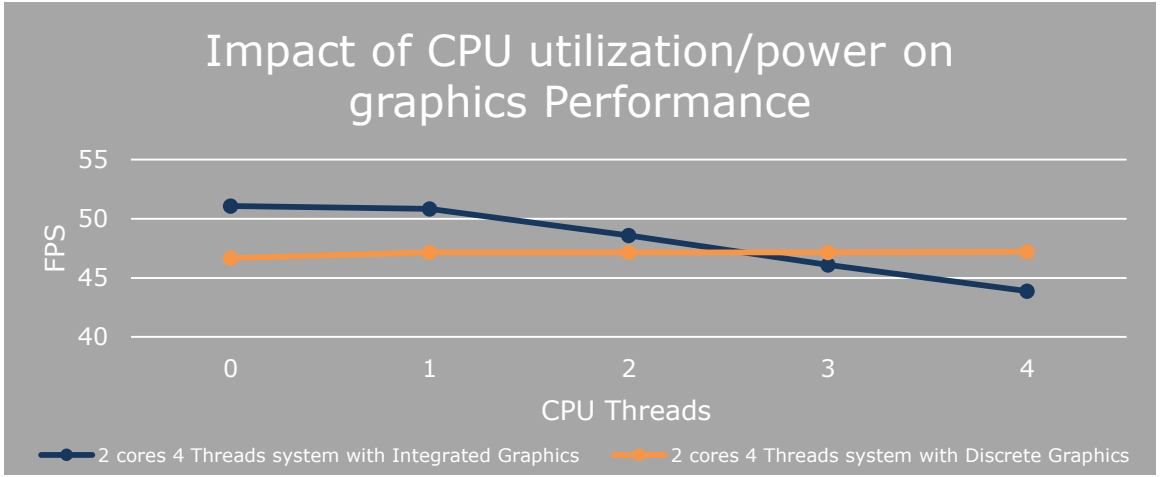
"Will destroy my battery life, though"

*"Never seen a game **drain battery** like this. Even my 6s loses 10% in 20 minutes"*

*"**Kills battery** and limits where one can play."*

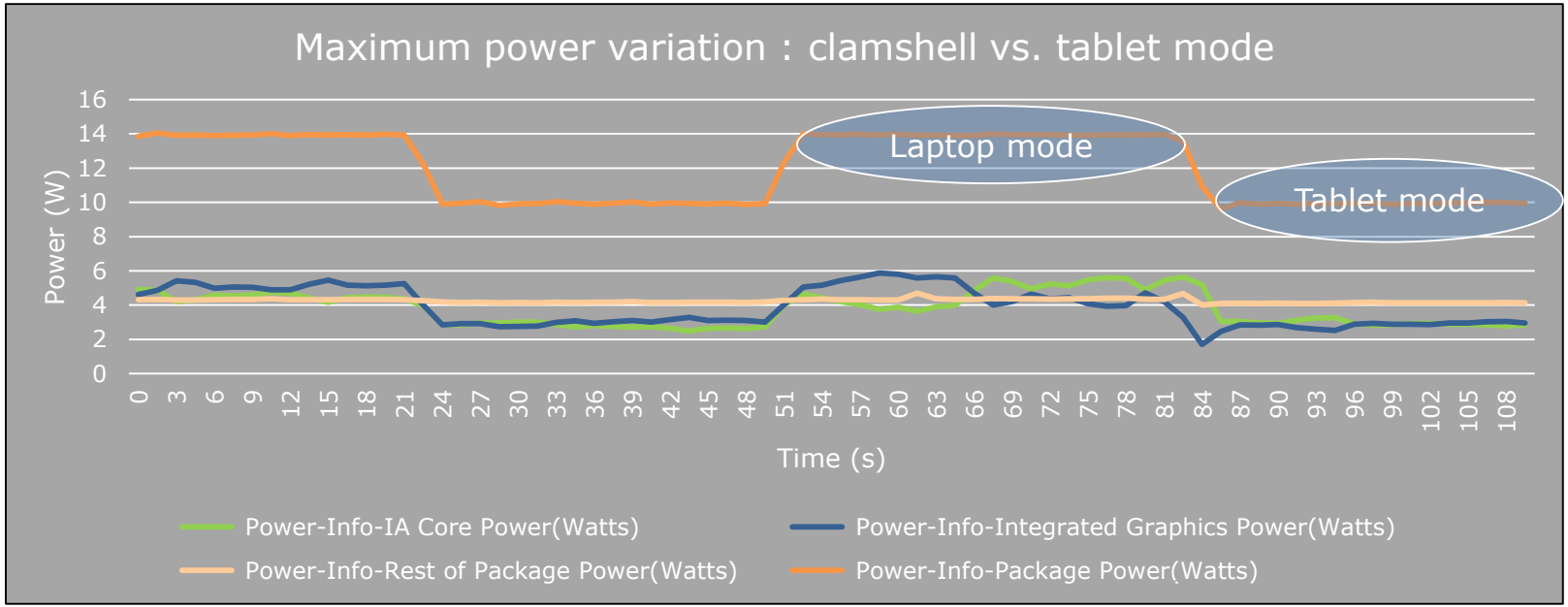
Why power matters

- Integrated CPU and GPU share the same power envelope:
  - Even if a game is 100% GPU limited, CPU workload can impact performance
- Case study:
  - Synthetic workload
  - GPU limited
  - Running some heavy kernels on the CPU

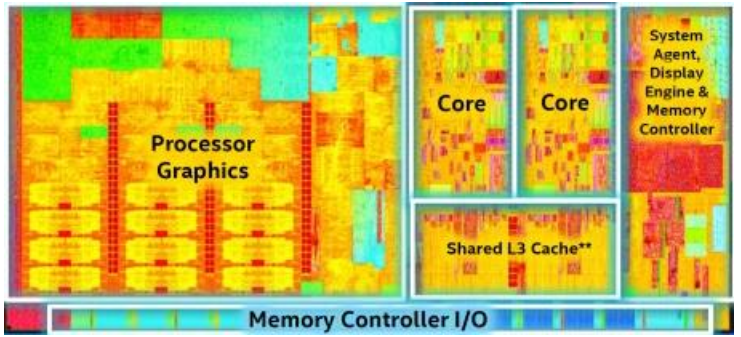


Why power matters

- Varying power behaviors:



ACPI in a nutshell



- Each SoC components has sleep states
  - The deeper the component sleeps, the more power is saved
- When active, components have Performance States
  - Each P-State has a given voltage and frequency
- Package C-State is determined by the higher of the cores
  - Any awake component will keep the package awake

Best practices: Adapt to your platform

- Numerous processors and platforms
  - Differences can impact performance
  - Take these factors into account !

- Profile the platform at install time
  - GPU detect / Run some short game snippet
  - Select the optimal tradeoff performance/settings

Best practices: Cap the frame rate

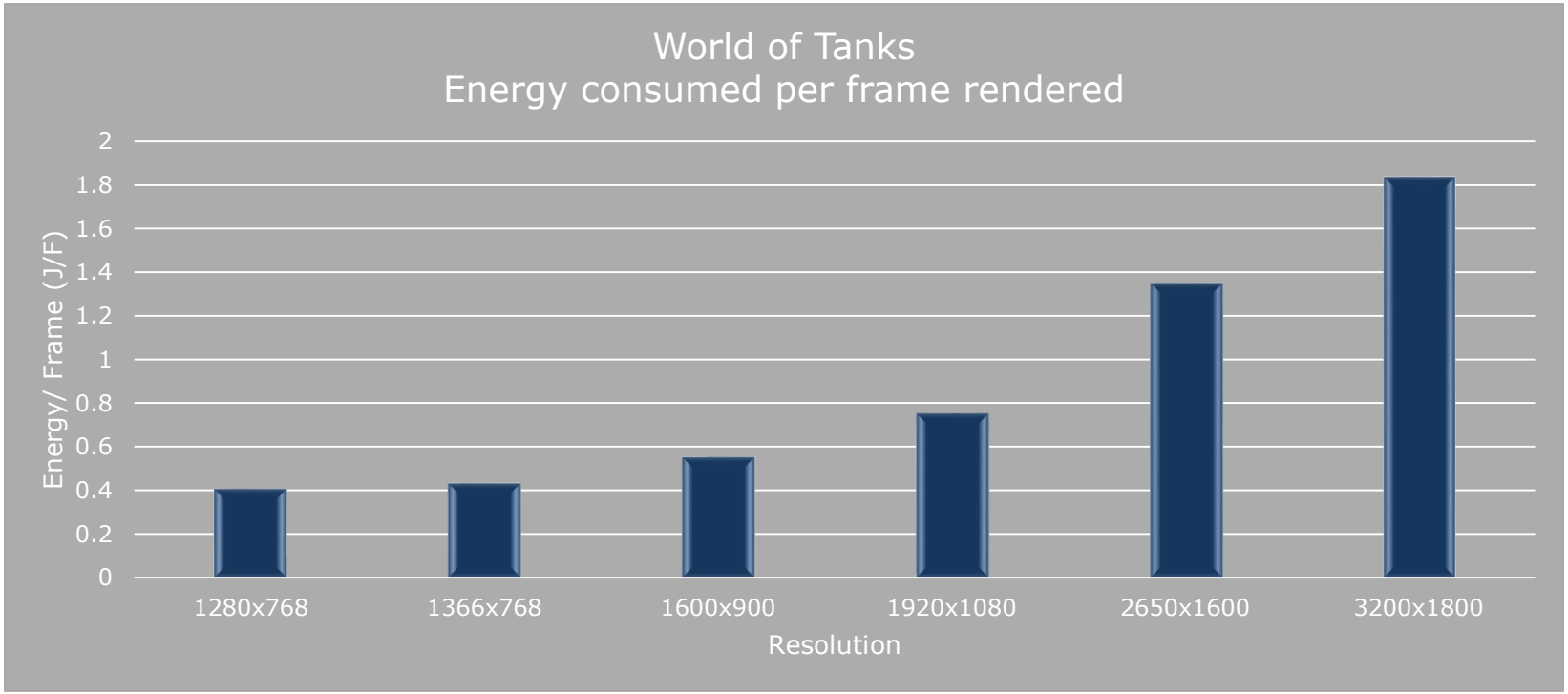
- Easiest way to save power, and also most efficient
  - DirectX - HRESULT IDXGISwapChain::Present(SyncInterval, Flags);
  - OpenGL ES - EGLBoolean eglSwapInterval(EGLDisplay display, EGLint interval);
- 2x less work

	Interval = 2	Interval = 1	
	FPS	FPS	Battery Life increase
Cut The Rope 2	30.2	40.3	111%
Epic Citadel	31.1	50.7	179%
Intel Fast Blur	31.0	63.0	133%
Zombie Tsunami	30.1	60.4	136%

Source : Internal Intel test results on a Intel® Atom™ Processor Z3745 reference design

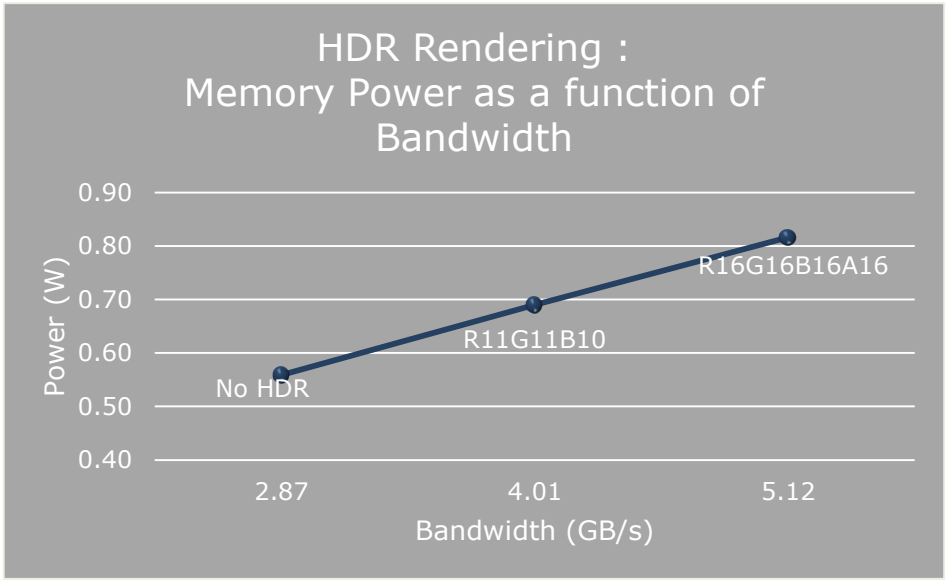
- Cap frame rates for menus/static scenes even lower

Best practices: Find a reasonable resolution



Best practices: Keep bandwidth under control

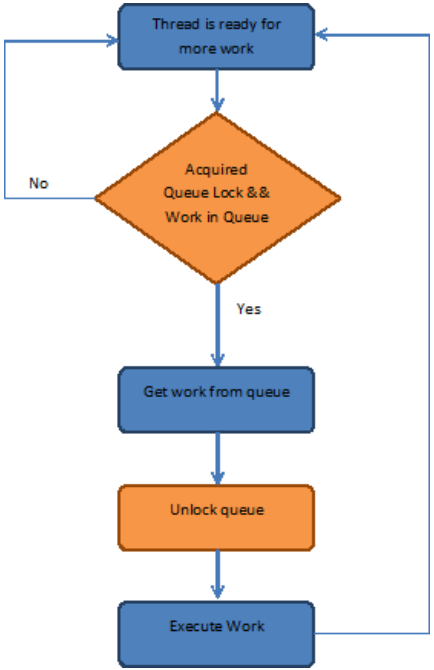
- 2560x1600x60 fps => 1GB/sec
- Design points for some of the tablets in the marketplace is 2-3W
- Changing RT format can help save 15-20% power



Source : Internal Intel® test results on a core i7-4610Y w/ 2x2GB LPDDR3 1600

Best practices: Careful with spin loops

- Widely used:
  - Reduce input latency
  - Thread pools
- Prevents CPU from sleeping!
  - Avoid using them as much as possible
  - If really necessary, use the pause instruction



https://software.intel.com/en-us/articles/benefitting-power-and-performance-sleep-loops

Best practices: A lot more…

- Handle loss of focus
- Color buffer clears
- 2D game optimizations using depth
- Reduce CPU activity
- Use of next gen APIs

Lego Minifigures: Introduction

- Funcom's previous games…
  - Designed for a mature audience
  - Advanced graphics technologies
  - Mainstream was not a target
- Lego Minifigures
  - Targeting kids
  - New renderer:
    - Designed for mainstream …
    - … but with high end effects!
  - Scalability is key

Lego Minifigures: Power saving mode

- Cap the framerate to 30FPS
- Simpler lighting and shadows
- No anisotropic filtering
- Post process FXs
- Disable antialiasing

Lego Minifigures: Frame capping

- Vsync vs. Sleep

- Results:

	Before	After	Power Saved (%)
System Power (W)	16.7	12.3	26%
Soc Power (W)	10.1	6.6	35%
Memory Power (W)	1.4	1.1	25%

Source : Internal Intel® test results on a core i7-4610Y w/ 2x2GB LPDDR3 1600

Lego Minifigures: Shadows Quality

- Optimization:
  - Lower resolution shadow-maps
  - Simple filtering
  - Disabled AVSM
- Results:

	Before	After	Power Saved (%)
System Power (W)	12.3	11.8	4%
SoC Power (W)	6.6	6.1	8%
Memory Power (W)	1.1	1.1	1%

Source : Internal Intel® test results on a core i7-4610Y w/ 2x2GB LPDDR3 1600

Lego Minifigures: Shadows quality

Lego Minifigures: Lighting and shadows

- Optimizations performed:
  - Still deferred – but simplified – lighting
  - Render only dynamic objects to shadow map
  - Disable HBAO
- Results:

	Before	After	Power Saved (%)
System Power (W)	11.8	10.6	10%
SoC Power (W)	6.1	5.1	16%
Memory Power (W)	1.1	1.0	6%

Source : Internal Intel® test results on a core i7-4610Y w/ 2x2GB LPDDR3 1600

Lego Minifigures: Lighting and shadows

Lego Minifigures: Disable PP effects

- Disabled depth of field
- Disabled god rays
- Disabled CMAA

- Results :

	Before	After	Power Saved (%)
System Power (W)	10.6	10.3	3%
SoC Power (W)	5.1	4.8	5%
Memory Power (W)	1.0	0.9	9%

Source : Internal Intel® test results on a core i7-4610Y w/ 2x2GB LPDDR3 1600

Lego Minifigures: Disable pp effects

Lego Minifigures: Final overview

	Power Saving Mode	Battery life	Battery life Increase
Asus TP 300LD (NVidia GT820M)	OFF	01:53:01	79%
	ON	03:22:04	
System with HD Graphics 5300	OFF	01:53:02	103%
	ON	03:49:04	

Source : Internal battery rundown tests

Conclusion

- Being power friendly: as easy as 1,2,3 !
- Users care about battery life: let them choose!
- Power optimization is performance optimization
- Focused optimizations can bring you terrific battery gain

 @acohade1

 antoine.cohade@intel.com

Useful links

- GPU Detect : https://software.intel.com/en-us/vcsource/samples/gpu-detect
- Loops with Pause : https://software.intel.com/en-us/articles/benefitting-power-and-performance-sleep-loops
- Power Explorer: https://software.intel.com/en-us/blogs/2013/10/29/power-explorer
- DX12 Siggraph Demo: http://blogs.msdn.com/b/directx/archive/2014/08/13/directx-12-high-performance-and-high-power-savings.aspxt