GPU hardware detection for automatic configuration of game quality/performance settings

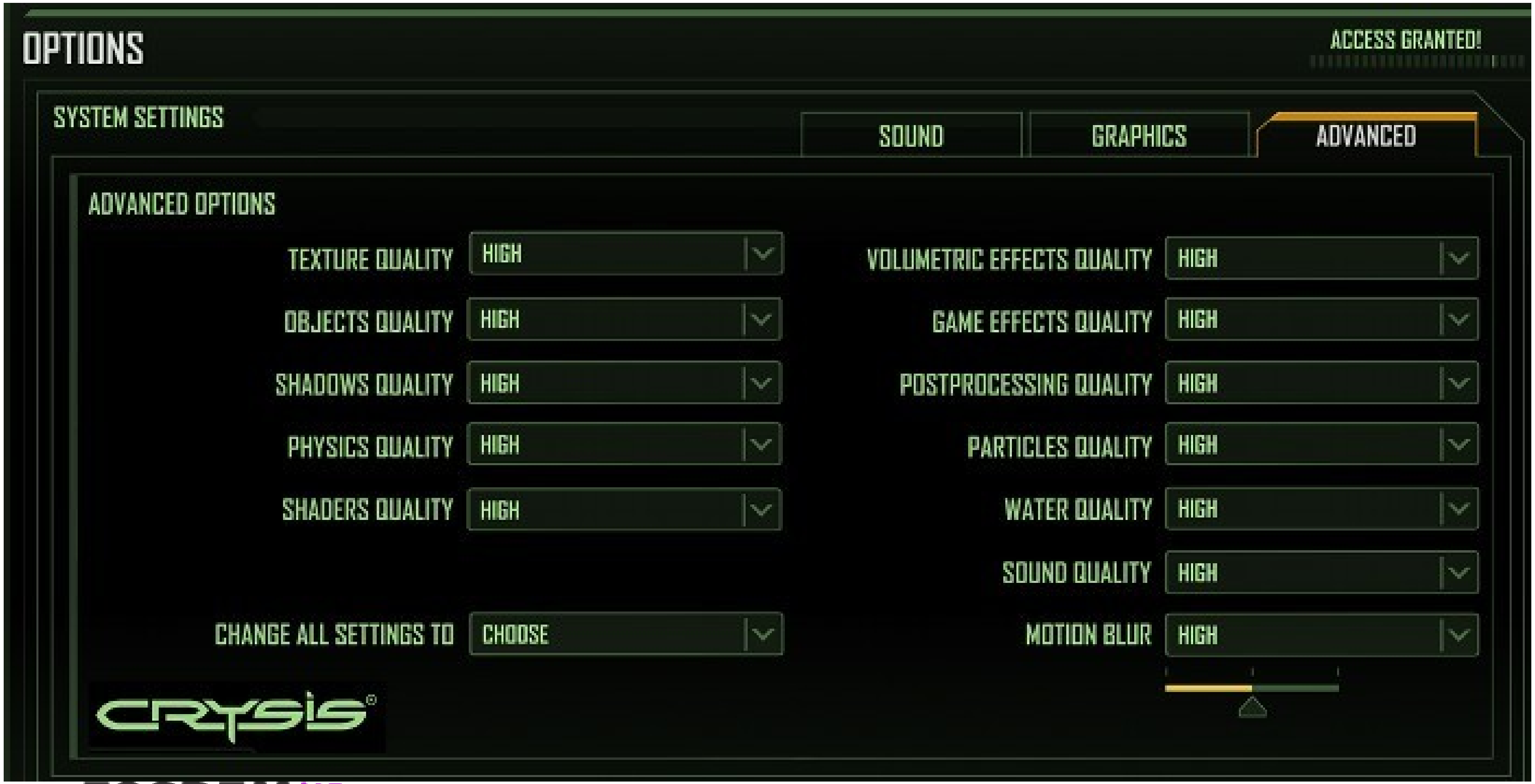
Ian Romanick
ian.d.romanick@intel.com

2-February-2013

Why?

- Quality performance trade-offs
  - Texture sizing
  - Shader quality
  - etc.
  - Pick *initial settings* for out-of-the-box exerpience
- Bug / hardware limitation work arounds
  - Black-list drivers with known security bugs

Why?


OPTIONS

ACCESS GRANTED!

SYSTEM SETTINGS

SOUND | GRAPHICS | ADVANCED

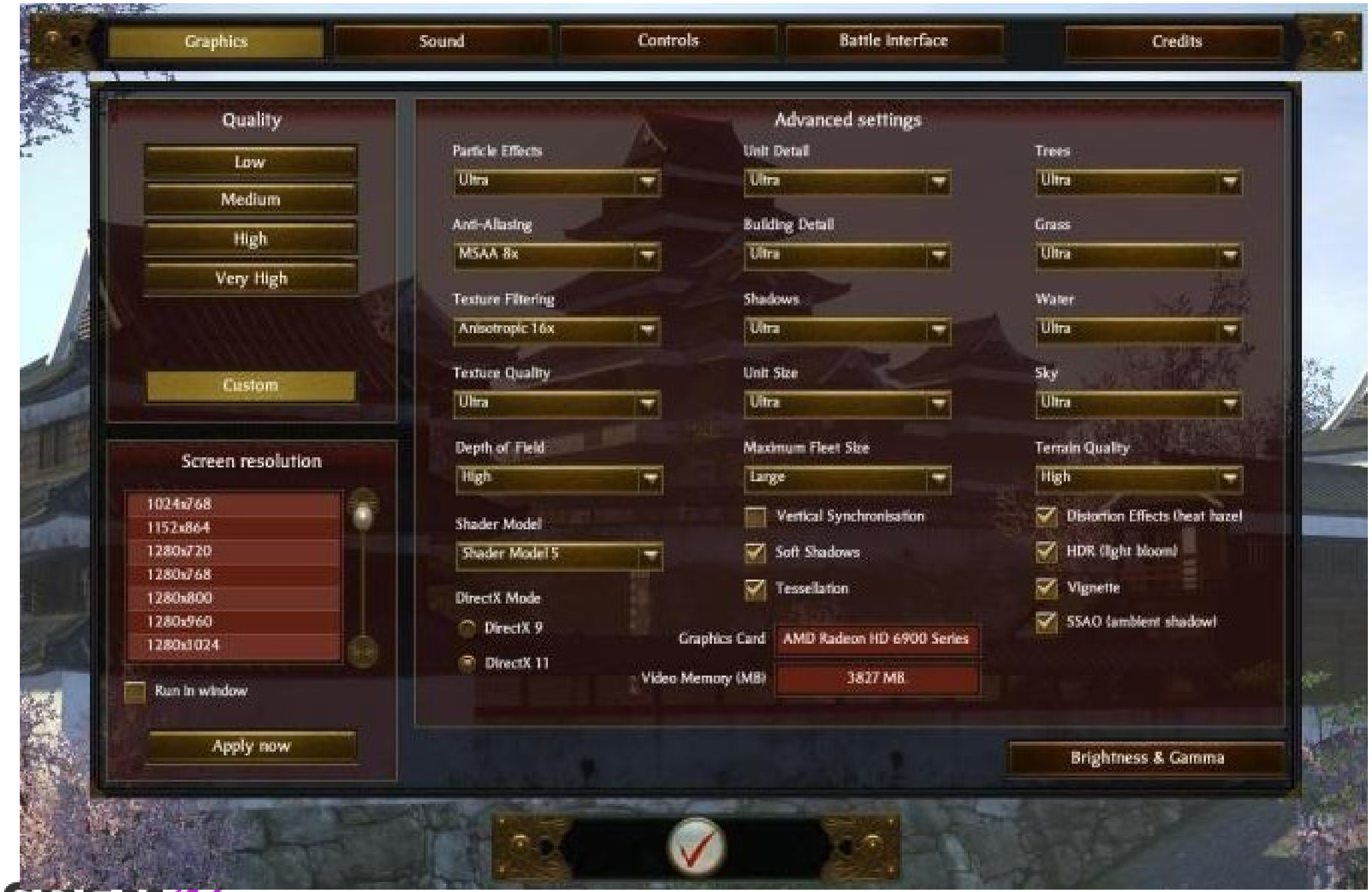
ADVANCED OPTIONS

TEXTURE QUALITY	HIGH	VOLUMETRIC EFFECTS QUALITY	HIGH
OBJECTS QUALITY	HIGH	GAME EFFECTS QUALITY	HIGH
SHADOWS QUALITY	HIGH	POSTPROCESSING QUALITY	HIGH
PHYSICS QUALITY	HIGH	PARTICLES QUALITY	HIGH
SHADERS QUALITY	HIGH	WATER QUALITY	HIGH
		SOUND QUALITY	HIGH
CHANGE ALL SETTINGS TO	CHOOSE	MOTION BLUR	HIGH

CRYSIS®

FOSDEM '13

Why?



Screen resolution

1024x768
1152x864
1280x720
1280x768
1280x800
1280x960
1280x1024

Run in window

Advanced settings

Particle Effects: Ultra	Unit Detail: Ultra	Trees: Ultra
Anti-Aliasing: MSAA 8x	Building Detail: Ultra	Grass: Ultra
Texture Filtering: Anisotropic 16x	Shadows: Ultra	Water: Ultra
Texture Quality: Ultra	Unit Size: Ultra	Sky: Ultra
Depth of Field: High	Maximum Fleet Size: Large	Terrain Quality: High
Shader Model: Shader Model 5	Vertical Synchronisation	✓ Distortion Effects (heat haze)
	✓ Soft Shadows	✓ HDR (light bloom)
	✓ Tessellation	✓ Vignette
		✓ SSAO (ambient shadow)

DirectX Mode: DirectX 9 [✓ DirectX 11]

Graphics Card: AMD Radeon HD 6900 Series

Video Memory (MB): 3827 MB

Hardware Detection

- Use libpci to find the display controller device

```
struct pci_access *pacc = pci_alloc();
pci_init(pacc);
pci_scan_bus(pacc);
for (struct pci_dev *dev = pacc->devices; dev != NULL; dev = dev->next) {
    pci_fill_info(dev, PCI_FILL_IDENT | PCI_FILL_CLASS);
    if (dev->device_class != 0x0300)
        continue;

    ...
}
pci_cleanup(pacc);
```

- Look up `vendor_id` and `device_id` in a table of quirks, default values, etc.

Driver Detection

- Query vendor, renderer, and version strings

```
GL_RENDERER
GL_VENDOR
GL_VERSION
```

- Format of these strings is implementation dependent
  - Search for key words like “Mesa”, “NVIDIA”, “ATI”, etc.
  - Driver version is *usually* somewhere in the `GL_VERSION` string
    - Mesa drivers report “3.1 Mesa 9.1-devel”
    - NVIDIA reports “3.3.0 NVIDIA 304.48”
    - AMD reports “3.3.11399 Compatibility Profile Context”

Memory Detection

- `GL_ATI_meminfo`
  - Query available memory in separate pools

```
GL_VBO_FREE_MEMORY_ATI
GL_TEXTURE_FREE_MEMORY_ATI
GL_RENDERBUFFER_FREE_MEMORY_ATI
```

  - Gives back total memory in pool, largest block in pool, total “auxiliary” memory free, and largest auxiliary block
  - http://www.opengl.org/registry/specs/ATI/meminfo.txt

Memory Detection

- `GL_NVX_gpu_memory_info`
  - Query overall memory availability

```
GL_GPU_MEMORY_INFO_DEDICATED_VIDMEM_NVX
GL_GPU_MEMORY_INFO_TOTAL_AVAILABLE_MEMORY_NVX
GL_GPU_MEMORY_INFO_CURRENT_AVAILABLE_VIDMEM_NVX
GL_GPU_MEMORY_INFO_EVICTION_COUNT_NVX
GL_GPU_MEMORY_INFO_EVICTED_MEMORY_NVX
```

  - http://developer.download.nvidia.com/opengl/specs/GL_NVX_gpu_memory_info.txt

Problems

- Each vendor provides information in a different way
  - Different format of `GL_VERSION` string, etc.
  - Different extensions for memory information
- Have to create a context to get most of the data
  - Annoying for apps that have an external configuration program
- Probing PCI information using external library just sucks
  - Fails on multi-GPU systems
  - Not even possible on Android

Other Platforms

- Apple's CGL has `CGLDescribeRenderer`
  - Call *before* creating context
  - Query `kCGLRPTextureMemoryMegabytes` for memory info
  - Query `kCGLPFARendererID` for device ID
    - Not the PCI ID, but can be used similarly
  - A lot of other queries also available
- Use I/O Kit to get PCI ID
  - If you really need it

Other Platforms

- Windows has `EnumDisplayDevices`
  - Gets the PCI information
- Use GL extensions to query memory info

TURE INTEL LINUX WIRELESS GUPNP KVM POKY LINUX KERNEL
OP YOCTO CONNMAN XEN OFONO
CS SYNCEVOLUTION SIMPLE FIRMWARE INTERFACE (SFI) ENTERPRISE SECURITY INFRASTRUCTURE

INTEL OPEN SOURCE
TECHNOLOGY CENTER