Khronos Presentations for Today

• **Khronos Ecosystem - including Augmented Reality**
  - Neil Trevett, Khronos President and NVIDIA Vice President of Mobile Content

• **Market Impact of Khronos Standards since 2000**
  - Dr. Jon Peddie, President JPR Research

• **Khronos KITE – Connecting Industry and Education**
  - Erik Noreke, OpenSL ES and OpenMAX AL Work Group Chair

• **Getting Involved in Khronos Chapters**
  - Angela Cheng, Khronos China Business Development
Open API Standards
Taiwan, June 2012

Neil Trevett
President, The Khronos Group
Vice President Mobile Content, NVIDIA
Topics

- Why open standards advance the industry
- How Khronos creates open API standards for silicon acceleration
- State of the art in visual computing APIs
- Vision and sensor processing for Augmented Reality
- Advanced hardware acceleration in HTML5
Why do we NEED Standards?

• Need standard INTERFACES for INTEROPERABILITY ...
  ... Widely adopted standard interfaces can grow market opportunity

• Interoperability enables compelling user experiences ...
  ... to be built cheaply enough to build a mass market

• Standards avoid fragmentation that adds no value ...
  ... Industries need to COOPERATE to build a market and then COMPETE

• The mobile industry is define by standards
  - GSM/EDGE, UMTS/HSPA, LTE, IEEE 802.11, Bluetooth, USB, HDMI ...
Khronos Interfaces Software to Silicon

- Khronos APIs define processor acceleration capabilities
  - Graphics, video, audio, compute, vision and sensor processing
- Driving market growth by expanding device capabilities
  - Enabling compelling applications and end-user experiences
APIs BY the Industry FOR the Industry

• Khronos standards have strong industry momentum
  - Shipping on billions of devices and multiple operating systems

• Khronos is OPEN for any company to join and participate
  - Standards are truly open – one company, one vote

• Khronos APIs define core device acceleration functionality
  - Low-level “Foundation” functionality needed on every platform

• They are FREE
  - No royalties
Over 100 members – any company worldwide is welcome to join

Board of Promoters
Khronos Working Group Process

Working Groups (WGs)
One working group per API

Academic Members
- Participation in WGs
Contributor Members
- Participation and vote in WGs
Promoter Members
- Participation and vote in WGs
- Board seat for strategy, budget and spec ratification

Members
Wider Industry

Conformance Tests and Adopters Program
Ratified Specifications
SDKs, Sample, Ref Cards and Man Pages

Adopters
Build conformant implementation and products

Developers
Develop applications using the APIs

Adopters
Build conformant implementation and products

Developers
Develop applications using the APIs

© Copyright Khronos Group 2012 | Page 10
Additional Khronos Initiatives

• **KITE**
  - Connecting Industry and Educators

• **Chapters**
  - Grass Roots Community for Khronos-related activities
  - Global network of like minded volunteers and participants – similar to ACM
  - Local networking, education and activities

WebGL Meet-up San Francisco Sep11
API Standards Evolution

**New API** technology first evolves on high-end platforms

**Desktop**
- OpenGL
- OpenMAX
- OpenCL

**Mobile**
- OpenSL ES
- Mobile is the new platform for apps innovation. Mobile APIs unlock hardware and conserve battery life

**Interop, Vision and Sensors**
- OpenVL
- Apps embrace mobility’s unique strengths and need complex, interoperating APIs with rich sensory inputs e.g. Augmented Reality

**Web**
- HTML5
- Diverse platforms – mobile, TV, embedded – mean HTML5 will become increasingly important as a universal app platform
Mobile Platform Innovation

- New platform capabilities being driven by SILICON and APIs

Console-Class 3D
Performance, Quality, Controllers and TV connectivity

Vision
Cameras as sensors, Computational Photography, Gesture Processing

Sensor Fusion
Devices become ‘magically’ context aware – location, usage, position

HTML5 and WebGL
Web Apps that can be discovered on the Net and run on any platform
3D has evolved over more than 30 years

‘Doom’ on a PC – 1993
*id Software*

‘Samaritan’ Real-time Demo on a PC – 2011
*Epic Unreal Engine*

http://www.youtube.com/watch?v=RSXyztq_0uM
OpenGL for Each Hardware Generation

1. X
   - Fixed Function

2. X
   - Vertex and Fragment Shaders

3. X
   - Geometry Shaders

4. X
   - Tessellation and Compute

'Shape realism'

'Surface realism'
Accelerating OpenGL Innovation

Bringing state-of-the-art functionality to cross-platform graphics

2004 2005 2006 2007 2008 2009 2010 2011

DirectX 9.0c DirectX 10.0 DirectX 10.1 DirectX 11

OpenGL 2.0 OpenGL 2.1 OpenGL 3.0

DirectX 9.0c

DirectX 10.0

DirectX 10.1

DirectX 11

OpenGL 3.2

OpenGL 3.3/4.0

OpenGL 4.1

OpenGL 4.2
OpenGL ES – Mobile 3D

- OpenGL for embedded and mobile devices
  - Eliminates redundant and legacy desktop features
  - Adds mobile-friendly functionality
- OpenGL ES 2.0 – released March 2007
  - Fully programmable vertex and fragment shaders
- The dominant 3D API for mobile devices
  - Widely adopted for STB, DTV, automotive,…
  - Hundreds and hundreds of millions shipped
- Runs high-end content and engines
  - UE3, Unity, Unigine, Rage
OpenGL ES Deployment in Mobile

Use of 3D APIs in Mobile Devices
Source: Jon Peddie Research

OpenGL ES is the 3D API used in Android, iOS and almost every other mobile and embedded OS – other than Windows
Processor Parallelism

OpenCL is a programming framework for heterogeneous compute resources
OpenCL – Heterogeneous Computing

- Framework for programming diverse parallel computing resources in a system

- **Platform Layer API**
  - Query, select and initialize compute devices

- **Kernel Language Specification**
  - Subset of ISO C99 with language extensions

- **Runtime API**
  - Execute compute kernels multiple devices
  - Gather results

- **OpenCL has Embedded profile**
  - No need for a separate “ES” spec
OpenCL Working Group Members

- Diverse industry participation – many industry experts
  - Processor vendors, system OEMs, middleware vendors, application developers
  - Academia and research labs, FPGA vendors
- NVIDIA is chair, Apple is specification editor
OpenCL Milestones

• Six months from proposal to released OpenCL 1.0 specification
  - Due to a strong initial proposal and a shared commercial incentive

• Multiple conformant implementations shipping on desktop
  - For CPUs and GPUs on multiple OS

• 18 month cadence between releases
  - Backwards compatibility protects software investment

- OpenCL 1.0 released – conformance tests released Dec08
- OpenCL 1.1 Specification and conformance tests released Jun10
- OpenCL 1.2 Specification and conformance tests released Nov11
- OpenCL on mobile platforms begin to ship 2012
OpenCL Roadmap

OpenCL-HLM (High Level Model)
Exploring high-level programming model, unifying host and device execution environments through language syntax for increased usability and broader optimization opportunities

Long-term Core Roadmap
Exploring enhanced memory and execution model flexibility to catalyze and expose emerging hardware capabilities

WebCL
Bring parallel computation to the Web through a JavaScript binding to OpenCL

OpenCL-SPIR (Standard Parallel Intermediate Representation)
Exploring low-level Intermediate Representation for code obfuscation/security and to provide target back-end for alternative high-level languages
High Level GPU Compute Initiatives

- **OpenACC**: Directives for accelerators
- **OpenMP**: C/C++ Compiler
  - Compile directives to steer offload
  - of code sections to accelerators
- **OpenCL**: Offload to data-parallel hardware through C++ syntax/compiler extensions
- **HLM**: Offload to data-parallel hardware through C++ syntax/compiler extensions
- **CUDA**: C/C++ compilers and libraries for CPU/GPU programming
- **RenderScript**: Express data parallel workloads
  - in Java and convert byte code into OpenCL for GPU offload
- **Aparapi**: C-level 3D rendering and compute APIs for offload from Dalvik applications with JIT compilation for device portability
- **Microsoft**: C++ Accelerated
  - Massive Parallelism
  - (C++ AMP)
- **Google**: The Portland Group

© Copyright Khronos Group 2012 | Page 24
Visual-based Augmented Reality

- **Camera Tracking**: Camera images used to track the camera’s location and orientation.
- **Other on-device Sensors**: Additional sensors used to enhance tracking.
- **Camera video stream sent to the compositor**: The camera stream is sent to a compositor for processing.
- **3D Augmentation Rendering**: 3D augmentations composited with the video stream for seamless integration.

**Diagram Components**
- **Camera**: Begins the process by capturing the environment.
- **Video Stream**: Transmitted to the compositor for augmentation.
- **3D Augmentations**: Added to create a visual experience.
- **Display**: Shows the final augmented reality view.
AR = Input AND Output Processing

- Augmented Reality is uniquely challenging for a mobile SOC
- Needs sophisticated INPUT AND OUTPUT processing
- *In perfect harmony*

Detecting user gestures and actions

Sensing and tracking the context and the scene around the user

Augmented Reality Application

Generating 3D Graphical Augmentations

Compositing real and synthetic elements to delight and inform
OpenMAX AL – Streaming Media

- Enables key image, camera and video use cases
  - Allows optimal hardware acceleration with app portability
- Create Media Objects to process images and video with AV sync
  - Connect to variety of input and output objects to PLAY and RECORD media

- Advanced image capture and photography
- HD content playback with robust DRM
- HD video teleconferencing
- Augmented Reality
Possible Camera Extensions for AR

- **Query camera information**
  - Focal length (fx, fy), principal point (cx, cy), skew (s), image resolution (h, w)
  - Spatial information of how cameras and sensors are placed on device
  - Calibration and lens distortion

- **ROI extraction**
  - From wide angle and fish-eye lenses

- **FCAM++ - Extensive exposure parameters in single or burst mode**
  - Shutter, aperture, ISO, white balance, frame rate, focus modes, resolution
  - Synchronization with other system sensors

- **Data output format control**
  - Grayscale, RGB(A), YUV
  - Access to the raw data e.g. Bayer pattern
EGLStream – Video/Graphics Interop

OpenMAX AL Media Player is the EGLStream “Producer” and controls production of frames.

EGLStreams enables and hides details of video frame transport. Enables multiple buffering modes for different uses cases eg: FIFO and explicit latch/release.

OpenGL ES GL_TEXTURE_EXTERNAL is the EGLStream “Consumer” and converts video format into RGB OpenGL ES texture.

Camera File URL Etc. → OpenMAX AL MEDIA PLAYER Object → EGLStream → OpenGL ES GL_TEXTURE_EXTERNAL
OpenSL ES – Advanced Audio

• **OpenSL ES does for audio what OpenGL ES does for graphics**
  - Advanced audio functionality from simple playback to full 3D positional audio

• **Object-based native audio API for simplicity and high performance**
  - Same object framework as OpenMAX AL
  - Reduces development time

• **Attractive alternative to open source frameworks**
  - Tightly defined specification with full conformance tests
  - Robust application portability across platforms and OS
Market Demand for Sensor Fusion API

- Innovative use of growing sensor diversity
- Effective use of multiple interoperating sensors in one app
- PORTABLE apps need to be isolated from sensor details
- Application developers do not wish to be Sensor Fusion experts
- Do NOT force the application developer to access individual sensors (unlike almost all other sensor APIs)
- High-level API enables sensor vendors to drive and deliver competitive sensor fusion innovation
Portable Access to Sensor Fusion

Apps request semantic sensor information
StreamInput defines possible requests, e.g. “Provide Skeleton Position” “Am I in an elevator?”

Advanced Sensors Everywhere
RGB and depth cameras, multi-axis motion/position, touch and gestures, microphones, wireless controllers, haptics keyboards, mice, track pads

Apps Need Sophisticated Access to Sensor Data
Without coding to specific sensor hardware

Processing graph provides sensor data stream
Utilizes optimized, smart, sensor middleware
Apps can gain ‘magical’ situational awareness
Current StreamInput Participants

- Aiming for specification release in 2012
StreamInput Working Group Status

• Working Group announced August 2011
  - Steady growth - additional participants always welcome

• Prioritizing semantics and node interfaces for key sensor types
  - Motion tracking devices (accelerometers, gyroscopes, etc.)
  - Cameras (Depth, RGB)
  - Mice, keyboards, joysticks, touch surfaces, audio in/out

• Working Group received high-quality starting-point proposals
  - PrimeSense (OpenNI), Softkinetic (IISU), Sensor Platforms, TransGaming

• Expecting stable spec by Q3’2012
  - Can be implemented natively or layered
  - over existing OS APIs for fast adoption
OpenVL

- **Vision Hardware Acceleration Layer**
  - Enable hardware vendors to implement accelerated imaging and vision algorithms

- **OpenVL can be used by high-level libraries or applications directly**
  - Primary focus on enabling real-time vision apps on mobile and embedded systems

- **OpenCV is widely used open source library for vision projects**
  - Future version will leverage OpenVL

- **Working group aiming for stable draft spec in 2012**
Possible Implementation of Vision Stack

Semantics and fusion of camera and positional sensors

High-level computer vision library

OpenVL

Accelerated computer vision algorithms

Parallel computation
Example use of Khronos APIs in AR

- Positional Sensors
- Positional and GPS Sensor Data
- Computer Vision and Tracking
- Synchronization and sensor fusion
- OpenVL
- Camera Processing
  - Control Camera, Preprocess and generate video streams
- EGLStream
- Video stream to GPU
- Video TAP to CPU
- OpenMAX AL
- OpenVL
- 3D Rendering and Video Composition
- Audio Rendering
- Position and Tracking Semantics
- Application on CPU
- OpenSL ES

© Copyright Khronos Group 2012 | Page 37
HTML5 – Cross OS App Platform

- Increasing diversity of devices creates a demand for a true cross OS programming platform
- BUT need more than “more HTML”

Traditional Web-content

Rich Experiential Processing
- Multi-core CPUs
- Rich 2D and 3D GPU
- GPU Computing
- Multiple HD cameras
- Image and vision processing
- Video encode/decode
- Audio encode/decode
- Inertial and positional sensors

How can the Browser rapidly assimilate such diverse functionality?
Leveraging Proven Native APIs into HTML5

- Leverage native API investments into the Web
  - Faster API development and deployment
  - Familiar foundation reduces developer learning curve

- Khronos and W3C creating close liaison
  - Multiple potential joint projects

Native APIs shipping or working group underway
JavaScript API shipping or working group underway
Possible future JavaScript APIs
WebGL – 3D on the Web – No Plug-in!

- Historic opportunity to bring accelerated 3D graphics to web
  - WebGL defines JavaScript binding to OpenGL ES 2.0

- Leveraging HTML 5 and uses <canvas> element
  - Enables a 3D context for the canvas

- WebGL 1.0 Released at GDC March 2011
  - Mozilla, Apple, Google and Opera working closely with GPU vendors
WebGL Implementation Anatomy

Content downloaded from the Web. Middleware can make WebGL accessible to non-expert 3D programmers.

Browser provides WebGL functionality alongside other HTML5 specs - no plug-in required.

OS Provided Drivers. WebGL on Windows can use Google Angle to create conformant OpenGL ES 2.0 over DX9.
Rich WebGL / HTML Interaction

• 3D is not trapped in a rectangular window
  - 3D can overlay and underlay HTML content
  - Easy to make 2D HTML HUDs or 3D user interfaces

• Strong ties with other advanced HTML5
  - WebGL can use HTML5 <video>
    or canvas as a texture

• Render HTML DOM sub-tree as WebGL texture
  - Mozilla and Google prototyping as extension
  - Supports user interaction when pages in 3D
Frameworks and Tools

- WebGL is deliberately low level to enable the full power and flexibility of OpenGL ES 2.0
- If you are not an expert 3D programmer – don’t panic!
- WebGL is perfect foundational layer for JavaScript middleware frameworks
- Lots of utilities and tools already appearing
WebGL – Being Used by Millions Every Day
WebGL Deployment

- **WebGL 1.0 Released at GDC March 2011**
  - Mozilla, Apple, Google and Opera working closely with GPU vendors
  - IE can be enabled with Chrome Frame
    [https://developers.google.com/chrome/chrome/chrome-frame/](https://developers.google.com/chrome/chrome/chrome-frame/)

- **Mobile WebGL beginning to ship – Firefox, Opera**
  - Pervasive mobile WebGL expected during 2012

---

![Browser Support Chart](http://caniuse.com/#search=webgl)

WebGL is not enabled by default in desktop Safari.
On iOS 5 WebGL is available to iAds
WebGL and Security

- **WebGL is Architecturally Secure**
  - Currently, NO known WebGL security issues
  - Impossible to access out-of-bounds or uninitialized memory
  - Use of cross-origin images are blocked without permission through CORS
  - Browsers maintaining black lists - used if unavoidable GPU driver bugs discovered

- **DoS attacks and GPU hardening**
  - Draw commands can run for a long time -> unresponsive system
    - Even without loops in shaders
  - WebGL working closely with GPU vendors to categorically fix this
  - Short term: mandate ARB_robustness and associated GPU watchdog timer
  - Longer term: GPU provides increasingly robust security and multi-tasking
Need for 3D Transmission Standard

- **Need efficient codecs for 3D assets to grow 3D market opportunities**
  - Approaching chaos in 3D data encodings used in WebGL browsers and apps
  - Binary encoding – can be used together with JSON and RESTful APIs

- **Possible key requirements for common 3D binary encoding**
  - Full-scene - Geometry, textures, materials, animations, physics etc.
  - Compression of textures and geometry with streaming support with LOD flexibility

- **MPEG doing work on this - can achieve 1:40 – 1:120 compression**
  - Progressive Streaming using 3D Mesh Coding (3DMC)
  - Bones Based Animation (BBA)

- **Khronos and MPEG exploring collaboration**

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>H.264</td>
<td>PNG/JPEG</td>
<td>?</td>
</tr>
<tr>
<td>Amazon</td>
<td>YouTube</td>
<td>Facebook</td>
<td>?</td>
</tr>
</tbody>
</table>
Declarative 3D for the Web

• Need to enable ‘non-expert’ web programmers with layers over WebGL
  - 10,000s of 3D programmers worldwide versus millions of web developers
  - Middleware and layered architectures play a vital role

• W3C Incubator for Declarative 3D
  - "easy way to add interactive high-level declarative 3D objects to the HTML-DOM”

• Bind 3D even closer into the browser stack
  - Use as much HTML5 machinery as possible – DOM, JavaScript, CSS
  - Focus on driving optimized WebGL/OpenGL ES 2.0 back-end
  - Use Typed Arrays and drive for optimal performance

Scenegraph  
Immediate

   Canvas   WebGL

2D  3D
WebCL – Parallel Computing for the Web

• **JavaScript bindings to OpenCL APIs**
  - Enables initiation of Kernels written in OpenCL C within the browser

• **Bindings stay close to the OpenCL standard**
  - Maximum flexibility to provide a foundation for higher-level middleware
  - Minimal language modifications for 100% security and app portability
    - E.g. Mapping of CL memory objects into host memory space is not supported

• **API definition underway – public draft just released**

• **Compelling use cases**
  - Physics engines for WebGL games
  - Image and video editing in browser
WebCL Demo

http://www.youtube.com/user/SamsungSISA#p/a/u/1/9Ttux1A-Nuc

WebCL for Hardware-Accelerated Web Applications

Advanced Browser Technology
Samsung R&D Center
San Jose, CA
Multi-platform Visual Computing Ecosystem

Desktop

- **OpenGL**
  - Direct interop. Buffers and events

Mobile

- **OpenGL ES**
  - Interoperate through **EGL**
  - Full or Embedded Profile

Web

- **WebGL**
  - Typed Arrays. Canvas and Video element interop

- **WebCL**
Web Apps versus Native Apps

• Mobile Apps have functional and aesthetic appeal
  - Beautiful, responsive, focused

• HTML5 with accelerated APIs can provide the same level of “App Appeal”
  - Highly interactive, rich visual design

• Using HTML5 to create ‘Web Apps’ has many advantages
  - Web app is searchable and discoverable through the web
  - Portable to any browser enabled system
  - Same code can run as app or as web page
  - Not a closed app store – no app store ‘tax’
## API Adoption

### Mobile Operating Systems

<table>
<thead>
<tr>
<th>Installed Bases (millions of US users)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
</tr>
<tr>
<td>iPhone</td>
</tr>
<tr>
<td>BlackBerry</td>
</tr>
<tr>
<td>Microsoft Other</td>
</tr>
</tbody>
</table>

### OpenGL
- **OpenGL ES 2.0**
  - Shipping - Android 2.2
- **OpenGL ES 2.0**
  - Shipping - iOS
- **OpenGL 3.2**
  - on MacOS
- **OpenGL ES 2.0**
  - on iOS
- **OpenCL 1.1**
  - on MacOS

### OpenSL ES
- **OpenSL ES 1.0**
  - Shipping – Android 2.3

### OpenMAX AL
- **OpenMAX AL 1.0**
  - Shipping - Android 4.0

### EGL
- **EGL 1.4**
  - Shipping under SDK -> NDK

### WebGL
- Chrome on ICS will have WebGL.
- Opera and Firefox WebGL now
- Can enable on MacOS Safari
- iOS5 enables WebGL for iAds

### Microsoft WinRT/Metro
- Only Microsoft native APIs
- HTML5 but not yet WebGL
Extended Native APIs on Android

- Native APIs can be shipped as NDK extensions before Google Adoption
  - Do not break/change existing Google APIs
- Khronos APIs have strong momentum in the silicon community
  - Evangelize Google to adopt into standard platform
- Extended APIs can be used by:
  - Bundled apps, Market apps with API selection
  - Multiple APKs behind single multi-APK SKU

Evangelize Google and ISVs to adopt enabling APIs

E.g. video, camera, imaging, sensor and composition APIs
In Summary

• APIs are key to enable compelling applications on advanced hardware – APIs developed on high-end hardware are now enabling mobile devices
• APIs no longer exist alone – they interoperate and provide input AND output processing to form a complete platform for advanced content
• Significant cooperation underway between native and Web APIs to bring advanced visual computing to HTML5
• Khronos is driving open standards for hardware acceleration
  Participate, change the industry AND get the inside edge for your products!