WebGL and WebCL
3D Graphics and Compute on the Web

Neil Trevett
VP Mobile Content, NVIDIA
President, Khronos Group
Two WebGL Sessions Today

Industry ecosystem and standards overview

- Web and Mobile Ecosystem, Khronos and WebGL Overview
  - Neil Trevett – VP Mobile Content at NVIDIA, President of Khronos
- WebCL Overview
  - Tasneem Brutch, Sr. Staff Engineer, Samsung Electronics, WebCL Chair

Hands-On with WebGL

- WebGL technical overview
  - Ken Russell - Software Engineer, Chrome GPU team, WebGL Chair
- CubicVR overview, tutorial and demos
  - Bobby Richter, Creative Tech Lead, Web Made Movies, Mozilla Foundation
Topics for this Session

- What is 3D anyway?
- Khronos and open standard acceleration APIs
- The evolution of pervasive 3D on mobile platforms
- WebGL and HTML5
- 3D use cases beyond games - augmented reality
- Looking ahead to what’s next

WebGL Reference Cards at end of session!
What is Real-time 3D Graphics?

Computer graphics is the science and art of using computers to create and enjoy beautiful, interactive experiences. The processor that makes these amazing experiences possible is the GPU.
3D Pipeline Basics

- The art of “faking” a realistic looking scene or objects using heuristic techniques learned over the years
- Surfaces of objects are broken down into a grid of polygons
- The vertices of the polygons are located in 3D coordinate space - x,y,z
- Each vertex has a “material” - color and reflective properties
- The objects making up a scene are held in a database
Fundamental 3D Processing Stages

Operations on Vertices
- Geometry
  - Rasterization
    - Traversal
    - Transforms
    - Lighting
    - Rasterize

Operations on Pixels
- Color
- Clip
- Write

What objects are in the current scene?
Where are the polygons?
What color are the polygons?
What shape are they on the screen?
What color is each pixel?
Which pixels are visible?
Write the pixels to the framebuffer
Actual 3D Pipelines

OpenGL ES 1.x Fixed Function Pipeline

- API
- Primitive Processing
- Vertex Buffer Objects
- Transform and Lighting
- Primitives Assembly
- Texture Environment
- Colour Sum
- Fog
- Alpha Test
- Depth Stencil
- Colour Buffer Blend
- Dither
- Frame Buffer

OpenGL ES 2.0 Programmable Pipeline

- API
- Primitive Processing
- Vertex Buffer Objects
- Vertices Shader
- Primitives Assembly
- Fragment Shader
- Depth Stencil
- Colour Buffer Blend
- Dither
- Frame Buffer
3D evolving over more than 30 years

‘Doom’ 1993

‘Samaritan’ Real-time Demo - 2011
Khronos and Hardware APIs

- Khronos defines open, royalty-free standards to access graphics, media, compute and input hardware
- Khronos APIs are low-level - just above raw silicon - to create the “foundation” functionality needed on every platform
- Safe forum for industry cooperation
  - Open to any company to join
  - IP framework to protect members and industry
- By the industry for the industry
Khronos Family of Standards

Khronos creates royalty-free specifications to meet real market needs and helps drive industry adoption across multiple platforms.
Mobile Silicon Experiential Processing

- Cortex A9 Processor
- Cortex A9 Processor
- Image Processor
- HD Video Decode Processor
- HD Video Encode Processor
- Audio Processor
- 2D/3D Graphics Processor

- ARM 7

- Mobile device interfaces with HD Video, Audio, and Graphics processors.
Next Generation Mobile Processors

- **CPU**: 3X Performance
  - Quad Core up to 1.5GHz, NEON

- **POWER**: 20x Lower Standby Power
  - Due to ULP mode

- **VIDEO**: 4X Complexity
  - 1080i/p High Profile

- **GRAPHICS**: 3X Performance
  - 12 Core, Dual Pixel Pipe

- **MEMORY**: 3X bandwidth
  - DDR3L up to 1600 data rate

- **IMAGING**: Better noise reduction & color rendition
  - Two simultaneous streams

- **AUDIO**: HD Audio
  - 7.1 channel surround

- **STORAGE**: 2 - 6X faster
  - e.MMC 4.4 and SATA-II
Mobile Roadmap Acceleration

- TEGRA 2 (2011)
- KAL-EL (2012)
- WAYNE (2013)
- LOGAN (2013)
- STARK (2014)

Performance:
- 1x
- 5x (KAL-EL)
- 10x (WAYNE)
- 50x (LOGAN)
- 75x (STARK)

Production Devices:
- Core 2 Duo - Macbook Air

© Copyright Khronos Group, 2011 - Page 14
A New Era in Personal Computing

1990’s 2000’s 2010’s

PC Internet Mobile Computing
20 Years Faster to 100M Per Year

Cumulative Shipments

- iOS & Android
- MacOS & Windows

Units in Millions

Source: Gartner, Apple, NVIDIA
Mobile - Android Becoming Dominant OS

Annual Volume (Millions)

Source: Gartner, NVIDIA
Mobile Form Factor Innovation

Motorola Atrix Media Dock

Motorola Atrix Nettop

Sony S2

Asus Transformer
OpenGL Ecosystem – 3D Everywhere

Leading-edge functionality developed first on desktop

OpenGL ES 2.0 on desktop as subset of OpenGL 4.1 for mobile content flexibility - including native support for WebGL

Mobile functionality subset that is deployed on billions of devices

WebGL will drive new-generation security into OpenGL family

Pervasive OpenGL ES 2.0 availability enables Browser vendors to build 3D directly into HTML5
OpenGL ES Pervasiveness

- **OpenGL ES 1.1 - fixed-function pipeline**
  - Based on OpenGL 1.5
  - Vertex Arrays / Buffer Objects
  - Transform & Lighting
  - Multi-texturing (min 2 units)
  - Fixed-point & Floating-point profiles

- **OpenGL ES 2.0 - programmable pipeline**
  - Based on OpenGL 2.0
  - Adds vertex and fragment shader programming
  - Removes fixed function pipeline
  - Super-compact, efficient API
  - High level language (GLSL ES)
  - On-line or off-line compilation
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

 availability of OpenGL and OpenGL ES on almost every web-capable device
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

Content
JavaScript, HTML, CSS, ...

JavaScript Middleware

WebGL
HTML5
JavaScript
CSS

OpenGL ES 2.0
OpenGL
DX9/Angle

© Copyright Khronos Group, 2011 - Page 22
HTML5 Content Architecture

HTML content generated by layout engine ‘on page’

<video> tag

JavaScript drives interactivity for 2D and 3D graphics

<canvas> tag

Composition of off-screen buffers

Composition needs to be GPU accelerated

Video, Vector Graphics and 3D created off-screen buffers

CSS Layout and Transforms

All content passes through CSS layout

Lorem ipsum
WebGL and HTML Interaction

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

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

• 3D for core web UI - as well as content
  - Advanced transforms and special effects

• WebGL is democratization of 3D
  - Accessible, pervasive, enabling
  - Spawning amazing innovation
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 Deployment

- **Typed array 1.0 spec ratified by Khronos in May**
  - Supporting bulk data transfer between threads (workers)
  - Many use cases - background mesh loading, generation, deformation, Physics ...

- **1.0.1 release of WebGL spec and conformance suite imminent**
  - 100% robust stance on security
  - Fixing bugs in 1.0.0 conformance suite
  - Implementations will report `getcontext("webgl")` (not experimental)

- **Render HTML DOM sub-tree as texture prototype extension**
  - Support user interaction when in 3D
  - Mozilla and Google prototyping

http://caniuse.com/#search=webgl

Not enabled by default
WebGL Security

• Any new functionality in the browser increases exposure to attack
  - True since the beginning of the web – the new functionality becomes hardened

• ANY graphics in the browser need the GPU drivers to be hardened
  - HTML, Canvas, WebGL, Adobe Molehill, Silverlight 5 …

• WebGL is designed with security as the highest priority
  - Hardening is being strongly promoted and enabled

• Short term - browser vendors will maintain white and black lists
  - Compromised system can have WebGL disabled until mitigation developed

• Longer term - GPUs will provide increasingly robust security and tasking
  - GPU becoming a first-class computing platform alongside CPU
WebGL Security in the Press!

• Confusion in the industry as we start this hardening process
  - Shader programs cannot access general system resources or perform out of range memory access!

• Issues in the Press
  - Cross domain image access – timed loop attack
    - WebGL and HTML spec updates - mandating CORS for video, images and audio
    - Servers have to grant cross-domain access to media resources
  - General hardening
    - ARB_robustness extensions that provide additional protection being mandated
    - New robustness spec limits the side-effects of a GPU reset after a DOS attack
    - ANGLE shader validator improved; more improvements coming
Why Khronos?

• Unique forum where browser and GPU vendors can cooperate
• Opened process to enable cooperation with web community
Flash Stage 3D aka ‘Molehill’

- **GPU-friendly 3D ‘stage’ behind classic Flash graphics**
  - No interaction with classic Flash except as overlay to 3D

- **Healthy competition to WebGL**
  - Competition /S a good thing

- **Contrasting design vector to WebGL**
  - OpenGL ES 2.0 assembler

- **Portability at the cost of functionality**
  - No loops
  - Lowest common denominator

- **Competition will ensure WebGL keeps it’s eyes on the ball for security and portability**
3D is much more than ‘just’ games

- Augmented Reality is a great Lighthouse use case
  - Need consistent APIs
  - AND
  - Reliable interop between them

- Significant Functionality
  - Camera control
  - Image processing
  - Positional sensors
  - Parallel computing
  - Graphics rendering
  - Video/graphics composition
  - Positional Audio
  - 3D models over the network
Visual-based Augmented Reality

Camera video stream sent to the compositor

Camera images used to track the camera’s location and orientation

Camera Tracking

3D augmentations composited with video stream

3D Augmentation Rendering

Camera-to-scene transform locks the 3D rendering to the real world
APIs Needed for Visual AR

- **Camera Control and image pre-processing**
  - OpenMAX

- **Image and vision processing**
  - OpenCL

- **Positional Tracking**
  - Situational Sensor fusion and synchronization
  - StreamInput
  - Situational Sensors: GPS, Compass, Accelerometer, Gyro, Pressure, Temperature

- **Composition of augmentations and camera image**
  - Camera image available to GPU for composition
  - Composition of augmentations and camera image

- **Augmentation Rendering**
  - Camera-to-scene transform in application
  - Multi-core, multi-threaded programming - physics and games engines e.g. UNITY
  - Synchronized Spatial 3D Audio
  - OpenSL ES

- **Physical Interfaces for HMD**
  - Transmission of markers and 3D models
  - Augmentation 3D models and markers
  - hmd
  - COLLADA

- **2D / 3D Rendering**
  - Video/graphics composition
  - EGL

- **Situational Sensor fusion and synchronization**
- **Multi-core, multi-threaded programming**
- **StreamInput**
- **OpenCL**
- **OpenSL ES**
- **khronos group Industry Forum**
- **mPEG Industry Forum**

© Copyright Khronos Group, 2011 - Page 33
Processor Parallelism

**CPUs**
- Multiple cores driving performance increases
- Multi-processor programming - e.g. OpenMP

**GPUs**
- Increasingly general purpose data-parallel computing
- Graphics APIs and Shading Languages

**Emerging Intersection**

**Heterogeneous Computing**

OpenCL is a programming framework for heterogeneous compute resources
The BIG Idea behind OpenCL

- **OpenCL execution model** ...
  - Define N-dimensional computation domain
  - Execute a kernel at each point in computation domain

- **C Derivative to write kernels** - based on ISO C99
  - APIs to discover devices in a system and distribute work to them

- **Targeting many types of device**
  - GPUs, CPUs, DSPs, embedded systems, mobile phones.. Even FPGAs

### Traditional loops

```c
void trad_mul(int n,
    const float *a,
    const float *b,
    float *c)
{
    int i;
    for (i=0; i<n; i++)
        c[i] = a[i] * b[i]; }
```

### Data Parallel OpenCL

```c
kernel void dp_mul(global const float *a,
    global const float *b,
    global float *c)
{
    int id = get_global_id(0);
    c[id] = a[id] * b[id];
} // execute over "n" work-items
```
WebCL – Call for Participation

• At GDC Khronos announced new WebCL initiative
  - ‘To bring parallel computing to browsers’

• E.g. Physics engines to complement WebGL
  - Image and video editing in browser

• One possible direction is JavaScript binding to OpenCL
  - Security is top priority

• Khronos welcomes new members to help define and drive WebCL
  - info@khronos.org
Expanding HTML5 Capability

- Web is the most widespread cross-platform programming platform
  - HTML5 Canvas tag is opening the door to API innovation

- JavaScript is now a viable language for visual computing
  - Most native APIs enable local caching of geometry/configuration

- Opportunity to synergize between Web and native APIs
  - Increase leverage, reduce developer learning cycles

![Diagram of HTML5 capabilities]
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”
  - X3DOM (www.x3dom.org/) and XML3D (www.xml3d.org/)

- 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
It's Time for a 3D Delivery Format!

• Format soup - only 3D does not have any widely agreed delivery formats
  - COLLADA, KML, MPEG, VRML, JSON, X3D binary, PowerVR POD, GZIP etc. etc.

• Fundamental to a ‘3D on the Web’ infrastructure
  - Compression reduces delivery time
  - Streaming with LOD flexibility increases end-user responsiveness
  - Browsers, apps and silicon can implement native accelerated decoders
  - Enables widely accessible, efficient content repositories

• Khronos and MPEG starting discussions
  - Leverage MPEG-4 AFX…
  - Encode COLLADA full-scene - geometry, textures, materials, animations, physics…
  - Restful API to negotiate precise served content…

<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>
</tbody>
</table>
In Summary

- WebGL brings another vital piece of system capability into the HTML5 browser for web apps - 3D graphics
- WebGL is being deployed right now on PC - soon on mobile - and is being strongly supported by browser and GPU vendors
- WebGL is a low-level, secure technology - that can be used directly and will support a rich ecosystem of tools and frameworks
Get Involved!

- Engage with the WebGL working group on Khronos forums and mailing lists
- Let us know if you have news or links that Khronos can help highlight
  - info@khronos.org or edit the Wiki
- Local WebGL Meetups here in NYC

http://www.khronos.org/webgl/wiki/Main_Page