penGL ES

CSCI 4239/5239
Advanced Computer Graphics
Spring 2020

OpenGL ES

- OpenGL for Embedded Systems
  - Phones
  - Game consoles
  - Appliances
  - Avionics
  - Subsystems (e.g. browsers)
  - …
- Cross-platform, open, standard

OpenGL ES Momentum

- **The leading 3D rendering API for mobile and embedded devices**
  - Based on desktop OpenGL – but optimized for mobile / handheld devices
  - Removes redundancy & rarely used features - adds mobile-friendly data types
  - The power of OpenGL distilled into a much smaller package
- **OpenGL ES adopted by every major handset OS**
  - Pervasive mobile 3D is evolving fast
- **OpenGL ES has become the most widely deployed 3D API**
  - Used in diverse applications, devices and markets
  - Mobile phones, games consoles, personal navigation devices, personal media players automotive systems, settop boxes

What is it?

- OpenGL adapted for Embedded Systems
  - Less capable hardware
    - Limited memory
    - Limited processing power
    - Lower clock frequencies
  - Lower power consumption
    - Less heat dissipation
- Same familiar API
  - Subset of full OpenGL API
  - Powerful 3D graphics in your pocket

OpenGL Advantages

- Standard and Royalty Free
- Small footprint
- Low power consumption
- Seamless hardware acceleration
- Extensible and evolving
- Easy to use
- Well documented

Current Applications

- Mobile devices
  - iPhone/iPod/iPad
  - Android
- WebGL
  - Chrome, Firefox, Safari, Opera, IE11, ...
- Embeded systems
  - 3D displays

OpenGL ES 1.1

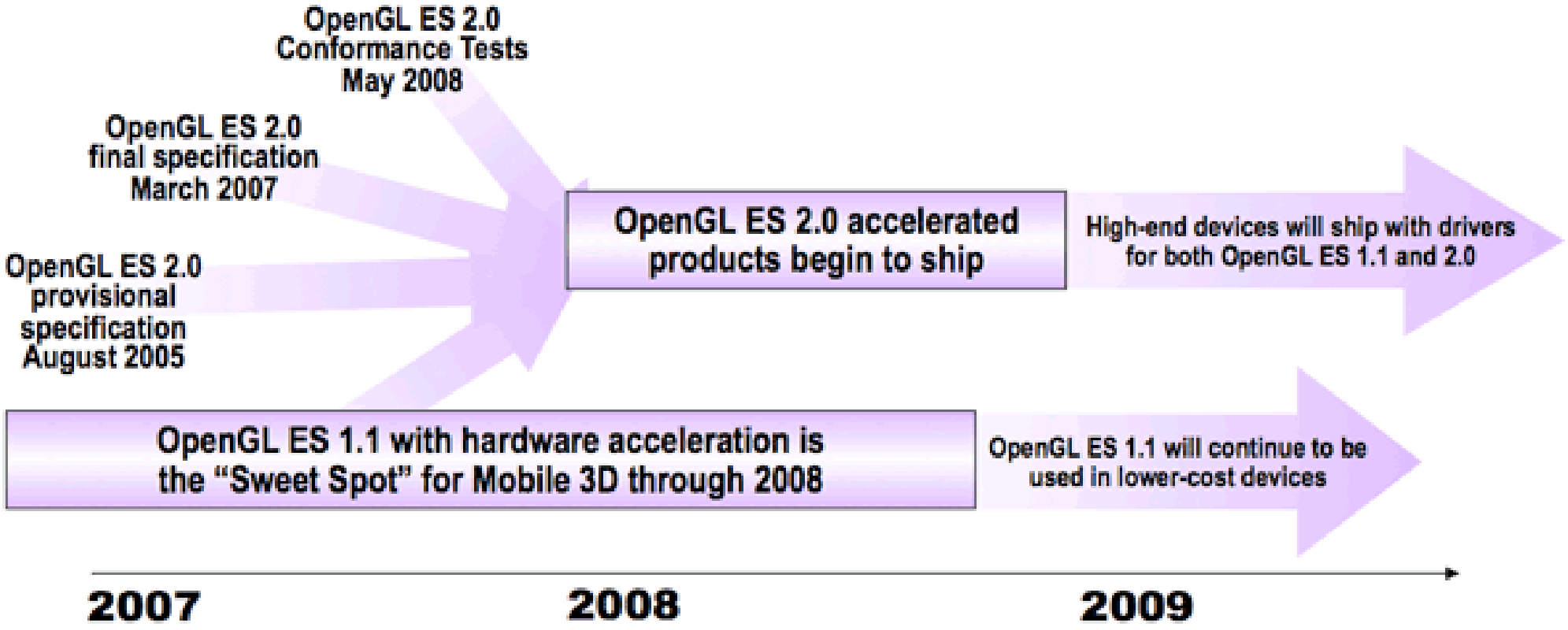
- Feature upgrade from OpenGL ES 1.0
- Defined relative to OpenGL 1.5
- Fixed pipeline (no shaders)
- Removes some functionality
  - No glBegin() … glEnd()
    - Replaced with glDrawArrays() & glDrawElements()
  - No GL_QUAD or GL_POLYGON
  - No display lists
- Still provides lighting, textures, etc.

OpenGL ES 2.0

- **Not backwards compatible with ES 1.1**
- Defined relative to OpenGL 2.0
- Shaders only (no fixed pipeline)
  - No lighting except in shaders
  - Textures only in shaders
- Removes transformation functions
  - No glRotate() glScale() glTranslate()
- OpenGL ES 3.0 adds feature upgrades

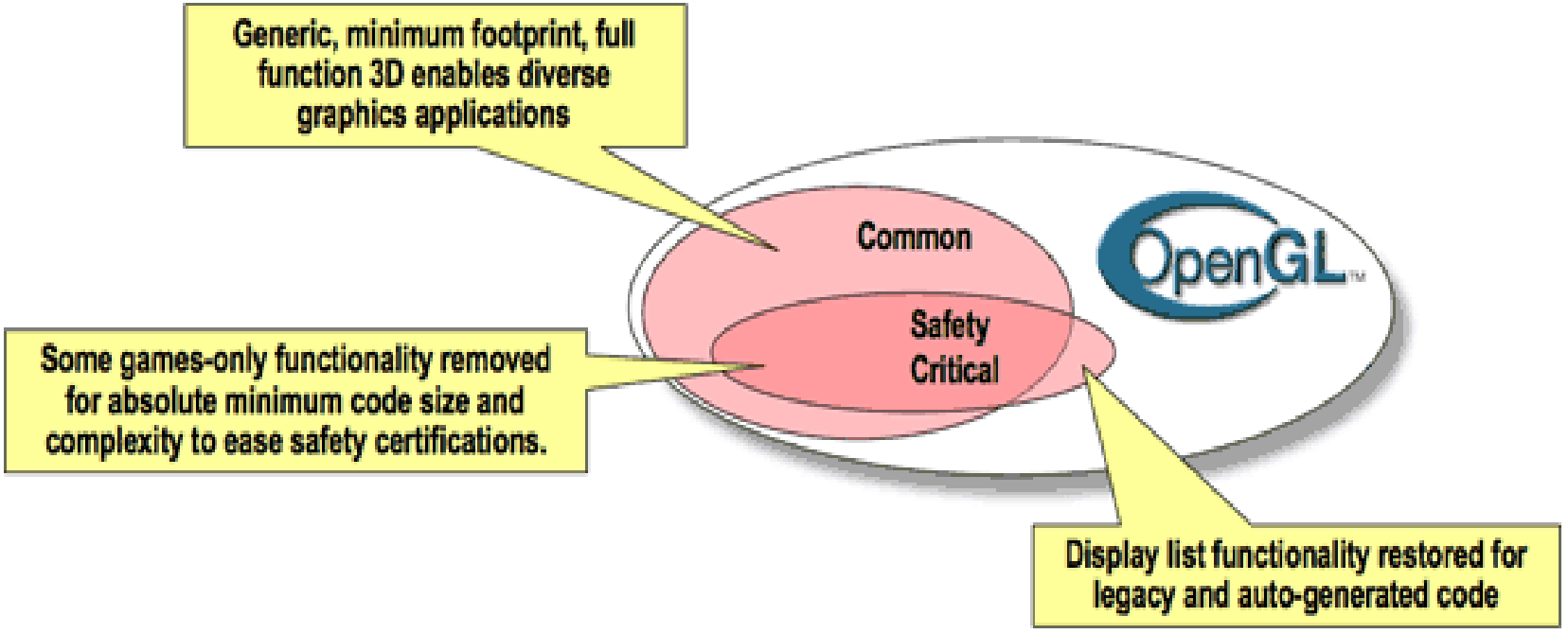
OpenGL ES Evolution

- **OpenGL ES 2.0 silicon implementations now shipping**
  - Shader-based graphics comes to mobile
  - Conformance tests shipping in May 2008
- **Listening carefully to implementation and developer feedback**
  - The determine next-generation requirements



OpenGL SC

- OpenGL for Safety Critical applications
  - Avionics
  - Automotive
  - Industrial
  - Medical
  - Military



OpenGL SC Features

- Starts with OpenGL ES 1.0
- Adds back some features
  - Begin/End
  - Display Lists
  - Some raster ops
  - Anti-aliasing
- Removes some features
  - Compressed textures
  - Multisampling
  - Fog
- Limits some features

EGL (Native Platform Interface)

- Performs some functions implemented by GLUT and SDL on the desktop
  - eglCreateWindowSurface()
  - eglSwapBuffers()
- Does not provide all the functionality provided by GLUT
  - User input
  - Redisplay

Apple iOS Devices

- Supports OpenGL ES 1.1 or 2.0
  - Newer devices support 1.1 AND 2.0
- User interface is Objective C
  - Links to C and C++ code
- Develop with Xcode on Mac only
- Emulator for all devices
  - Slower than native devices
  - Almost perfect emulation

Device Compatibility	Graphics Platform	OpenGL ES 2.0	OpenGL ES 1.1
iPod Touch	PowerVR MBX	No	Yes
iPod Touch (Second Generation)	PowerVR MBX	No	Yes
iPod Touch (Third Generation)	PowerVR SGX	Yes	Yes
iPod Touch (Fourth Generation)	PowerVR SGX	Yes	Yes
iPhone	PowerVR MBX	No	Yes
Phone 3G	PowerVR MBX	No	Yes
iPhone 3GS	PowerVR SGX	Yes	Yes
iPhone 3GS (China)	PowerVR SGX	Yes	Yes
iPhone 4	PowerVR SGX	Yes	Yes
iPad Wi-FI	PowerVR SGX	Yes	Yes
iPad Wi-Fi+3G	PowerVR SGX	Yes	Yes

Android Devices

- Supports OpenGL ES 1.1 or 2.0
  - Newer devices support 1.1 AND 2.0
  - Low end devices remains 1.1
- User interface is Java
  - Link to C/C++ code with JNI
- Develop with NDK
- Emulator for phones and tablets
  - Slower than native devices
  - Emulator (sorta) support OpenGL ES 2.0

Portable OpenGL ES Code

- Write the bulk of the code in C++
  - OpenGL ES 1.1 will run on all devices
  - OpenGL ES 2.0 will run on newer devices
- Write minimal code in interface language
  - Objective C – link to C/C++
  - Java – call C/C++ using JNI
- Qt 5 for iOS/Qt 5 for Android
  - Later builds are better

WebGL

- OpenGL ES 2.0 for the web
- Extends Javascript
- Operates on HTML5 canvas element
- Prohibits client side arrays
  - All vertex, normal, color, … must be stored in Vertex Buffer Object on video card
- Becoming more mainstream
  - Still a work in progress
  - Bleeding edge HTML & OpenGL

WebGL Platforms

- Supported by most browsers
  - Chromium
  - Firefox
  - Safari
  - Opera
  - Explorer
  - Edge
- Update to recent version

Assignment 5

- Create a scene that can be viewed in 3D using WebGL with lighting and textures
- Objects must be created by hand
  - I want you to get some experience using vertex buffer objects
  - May use CanvasMatrix library
  - May NOT use Three.js or similar libraries
- Explore features like buttons