Candera Device

DirectTextureImage External Buffers and New Formats

DirectTextureImage now supports application defined buffers to be provided for the texture data, as well as additionally supporting the 32 bit unpacked formats RGBA and BGRA.

This functionality can be used by setting the logical and/or physical address properties of DirectTextureImage using DirectTextureImage::SetLogicalAddress and DirectTextureImage::SetPhysicalAddress. The default behaviour of DirectTextureImage, where the buffers are allocated by the driver, is used if these properties are set to the default values. The addresses should be provided as a pointer array of four pointers in the case of logical addresses. Depending on the format, one or more pointers to provided buffers should be set and unused pointers in the array set to 0. In the case of physical addresses, these should be provided as an array of unsigned integers, where unused addresses are set to ~0U.

An additional function DirectTextureImage::Invalidate() has been added to the DirectTextureImage which should be called if the data in the provided buffers was modified.

OpenGlEs20 Device Package

Candera OpenGlEs20 Device Package has been added.

This is a generic device which supports 2Dover3D and takes device specific / native configuration from a Device Configuration Factory. Main purpose is quick setup of new OpenGL ES based platforms for evaluation or benchmarking.

Render API String in 3D Capabilities

DevicePackageDescriptor::RenderDevice3DCapabilities has been enhanced by a string indicating the render API, e.g. "OpenGL ES 3.0".

Revision #2
Created 2023-03-02 06:45:23 UTC by Kanai Tomoaki
Updated 2023-06-14 00:19:45 UTC by Kanai Tomoaki