Candera V3.0.4

Release Date: June 08, 2015

To keep backwards compatibility for any interface change in the public API the old interface has been kept and marked as deprecated. The deprecated interfaces will be removed in future versions. Compiler warnings will be generated when using a deprecated interface through the usage of Candera Macro CANDERA_DEPRECATED_3_0_4.

- Candera Engine 3D
- Candera Engine 2D
- Candera Base
- Candera System
- Candera Device
- TextEngine
- Migration Guide

Candera Engine 3D

Texture Level of Detail Parameters

The Texture class now supports setting a level of detail bias (offset or base level) as well as a lower and upper limit for the level of detail value. These are new parameters available in OpenGL ES 3.0 or higher, which allow control over which mipmap levels are sampled. The parameters are set through the RenderDevice class.

MSAA Offscreen Render Targets

Framebuffer objects now support setting the MsaaSamples property. If the device supports OpenGL ES 3.0 or higher, setting the property to anything larger than 1 will enable multisample anti-aliasing with the given number of samples for that framebuffer object.

NPOT Textures

NPOT (Non-Power-Of-Two) Textures are supported with OpenGL ES 3.0 API.

Vertex Geometry Modifier

VertexGeometry objects can be modified using the new class VertexGeometryModifier in conjuction with a VertexAccessor.

Occlusion Culling Camera Render Strategy

On OpenGL ES 3.0 platforms this camera render strategy uses hardware occlusion culling to improve performance. It only reaches its full potential when objects have been sorted front to back and occluded objects consist of lots of vertices, have expensive shaders, or both.

Render Order Performance Improvements

Nodes that are to be added to the default render order bins - Opaque and Transparent - are now added to these bins directly without expensive string comparisons in the Find function. The Find function was also modified to check the references of the default bin names before the string compares are done. Additionally a copy constructor was implemented for the RenderOrderBin class, allowing improvement of the copy constructor of RenderOrder.

Candera Engine 2D

Removed limitation of 256 layers for photoshop export/import

Return and parameters type for Node2D::SetRenderOrderRank, Node2D::GetRenderOrderRank and Node2D::GetEffectiveRenderOrderRank have been changed from Int8 and Int16 to Int.

LayoutingRectangle and BoundingBox calculation

LayoutingRectangle and BoundingBox are always computed from the first effect in the chain, if it is or contains a brush. If an effect is a combined effect it forwards the call to it's contained brush effect.

Added Computation of Perspective Warped Bounding Rectangle

A method for computing perspective warped bounding rectangle has been added to Math2D class.

See also:

Math2D::GetSurfaceAlignedPerspectiveBoundingRectangle().

Candera Base

Enums in DynamicProperties namespace renamed

The enumerations DynamicProperties::PropertyFlags::Enum and DynamicProperties::PropertyType::Enum were renamed to DynamicProperties::PropertyFlags and DynamicProperties::PropertyType. Additionally, the Generic PropertyType was renamed to GenericType.

Added 9 OpenGL ES3.0 texture formats

The following OpenGL ES 3.0 formats have been added to Candera::Bitmap

Bitmap::PixelFormat	**Bits per pixel**	**Renderable**	**Description**
EacCompressedUnsignedRPixelFormat	4		Compressed single channel unsigned alpha format
EacCompressedSignedRPixelFormat	4		Compressed single channel signed alpha format (can preserve 0 exactly)
EacCompressedUnsignedRGPixelFormat	8		Compressed two channel unsigned luminance alpha format
EacCompressedSignedRGPixelFormat	8		Compressed two channel signed luminance alpha format (can preserve 0 exactly)
SrgbUnsignedBytePixelFormat	24		sRGB
SrgbaUnsignedBytePixelFormat	32	X	sRGBA
RgbFloatingPointR11G11B10PixelFormat	32		Floating point format with 5 bit exponent and 5/6 bit mantissa
RgbFloatingPoint9E5FPixelFormat	32		Floating point format with shared 5 bit exponent and 9 bit mantissa
RgbaUnsignedIntRev10A2PixelFormat	32	X	RGBA with 10 bit per rgb, 2 bit alpha

Renderable means that the format can also be used as a render target.

Compressed single channel alpha formats deliver the best visual quality for transparency of all compressed formats using alpha. E.g., if the quality of transparency in a compressed texture is lacking, a combination of a compressed rgb texture together with a compressed single channel alpha texture should be favored over an uncompressed texture, since the 2 compressed textures still need less memory and bandwidth than an uncompressed one.

Candera System

Dynamic_Cast

Dynamic_Cast that uses pointer type as typename replaces the DynamicCast function that uses the referenced type as typename.

Candera Device

CLUT Bitmap Format

A new bitmap format, CLUT(color lookup table), has been added. This format offers compression by replacing the full RGB image with a palette having a limited number of entries and a color table which will contains references to the lookup index. In SceneComposer, the platform specific converter can be used to do the conversion of the bitmap into an indexed color format.

Added support for native invalid values for EGL display, surface and context types

Previous versions of Candera Device Package do not allow null values for EGL display, surface and context types. To overcome this limitation, support for native invalid values for EGL display, surface and context types has been added.

For this, three preprocessor defines were added in Candera:

- INVALID_NATIVE_EGL_CONTEXT
- INVALID_NATIVE_EGL_DISPLAY
- INVALID_NATIVE_EGL_WINDOW

In order to add custom invalid values for EGL display, surface and context types, one must override this defines with the desired values. For this, a header file must be created which first undefines and then defines the desired invalid values for INVALID_NATIVE_EGL_CONTEXT, INVALID_NATIVE_EGL_DISPLAY and INVALID_NATIVE_EGL_WINDOW. *EglInvalidValues.h* is the default header file used by Candera which can serve as a template for creating custom header files. In order to be take into account, this custom header file must be specified using the new CMake variable NATIVE_EGL_INVALID_VALUES_HEADER when the project is configured.

3DRendererAPI string

Candera::DevicePackageDescriptor::RenderDevice3DCapabilities structure has been extened with a string that specifies the 3D render engine API that it is used by the current or the target device.

Context caching

When RenderStateCaching is enabled, native calls to context activation (e.g. eglMakeCurrent) have to be avoided, because they will unsynchronize the cache. Activation of contexts can still be done via activation of Candera objects like: RenderTarget, ContextResourcePool or GduContext.

Perspective Viewport in 2D

The 2D Render Device has been enhanced to support setting a viewport used for perspective warping. This method can be used on platforms supporting such 2.5D effects.

See also:

RenderDevice2D::SetPerspectiveViewport()

TextEngine

Added new event triggered before bliting each glyph bitmap

A new method TextRendering::TextLayoutStrategy::OnGlyphProcessing has been added which is triggered before bliting each glyph bitmap. Default implementation has been added to keep backwards compatibility. This offers the possibility to skip the bliting of a glyph bitmap if needed.

For this, also the TextRendering::TextLayoutStrategy::Action enumeration has beed extended with a new *Break* value.

Migration Guide

Following an overview about interface changes between Candera V3.0.3 and Candera V3.0.4.

Description	Candera V3.0.3	Candera V3.0.4
Dynamic_Cast	`DerivedObject* derivedPtr = Dyn` `DerivedObject::SharedPointer de`	`DerivedObject* derivedPtr = Dynamic_Cast<Derived(` `DerivedObject::SharedPointer derivedPtr = Dynamic`