CGI Studio 3.0.4

- Candera V3.0.4
  - Candera Engine 3D
  - Candera Engine 2D
  - Candera Base
  - Candera System
  - Candera Device
  - TextEngine
  - Migration Guide
- Courier V3.0.4
- FeatStd V3.0.4
  - FeatStd Platform
  - FeatStd Util
- SceneComposer V3.0.4
  - Amber Platform Support
  - SceneComposer Usability Improvements

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

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`

Courier V3.0.4

3.0.4-1 Message Processing

A new Courier::ViewPlaceholderReqMsg message and the corresponding Courier::ViewPlaceholderResMsg have been added to Visualization messages.

These provide a better control of the loading/unloading of the render targets associated with a view. It gives the possibility to keep the render targets uploaded even if no corresponding view is loaded or activated.

3.0.4-2 Input Handling

A new Courier::InputHandling::InputContext class has been added, which is the generalize version of Courier::InputHandling::DisplayInputContext class without the dependency to Candera::Display. This gives the possibility to better handle the input for applications which don't handle a display (i.e. simple console/terminal applications)

Courier::InputHandling::DisplayInputContext has been reworked to inherit from Courier::InputHandling::InputContext and also handles the Candera::Display and Courier::InputHandling::DisplayConfiguration.

Courier::InputHandling::InputHandler and Courier::InputHandling::InputThread have been reworked to use the new Courier::InputHandling::InputContext class.

3.0.4-3 Widgets

New interface method Courier::FrameworkWidget::GetBindingsIterator has been added. For this also a new class Courier::FrameworkWidget::ConstBindingIterator has been implemented. This gives the possibility to iterate through the bindings of a widget. This can be further used to check various binding data (e.g. the target name to which the binding is bound).

FeatStd V3.0.4

FeatStd Platform

Thread extensions

The Thread class was extended with methods for:

- Retrieving the system unique thread Id for the thread that calls the static function: Thread::GetCurrentSystemId()
- Retrieving the system unique thread Id for the thread object on which the function is called: Thread::GetSystemId()
- Waiting for the thread associated with the thread object, on which the function is called, to finish: Thread::Join()

Lightweight Posix Critical Section

The generic critical section used for the posix platform was replaced with a lightweight, native implementation based on the pthread_mutex functions.

FeatStd Util

FeatStd::StringBuffer and StringBufferAppender

New FeatStd::StringBuffer interface class added for object to string conversion. Please refer to the API documentation for more details.

StringBufferAppender template and StringBuffer::AppendObject are used to append an object to the string buffer. These implementations are available only if FEATSTD_ENABLE_STRINGBUFFER_APPENDER is defined.

SceneComposer V3.0.4

Amber Platform Support

Amber Platform Support

The new platform - Amber - is strongly related to the new supported API - OpenGLES1.1. OpenGLES1.1 provides less features than OpenGLES3.0. The most important missing feature is the programmable pipeline - (Shader configurations). On Amber platform, one cannot create shaders. However, the platform is simulated over OpenGLES3.0 so there are some pre-defined shaders that emulate the specific 1.1 functionality.

In order to accommodate SceneComposer to Amber platform, several changes were done/are currently under development (Mantis 5393):

Shaders:

- Following Shader (substitutes) shall be exposed: Default Shader; BRDF Lighting Shader; Emission Texture Shader; Gradation Texture Shader.
- ShaderParamSetters (UniformSetters) remain and are required to set parameters of special effects, like BRDF.
- Shader Editor shall be read only. This allows at least to introspect usage of the effect parameters.
- Disallow creation and import of new shaders.
- Target Shader Compiler replaces effects with binary identifier in asset.

Textures:

- Amber supports two default or three BRDF textures
- SceneComposer shall allow three textures in general and throw warning if three textures are used with any other shader than BRDF
- If two default textures are used, internally the default blend mode is applied [Alpha blending with src*(srcAlpha) + dst*(1-srcAlpha)]
- No option to set (other) blend mode for multi-texturing.
- Tangients in VertexBuffer required for BRDF textures can be generated by SceneComposer

Render Mode:

- Alpha Blend Values are ignored, BackStencil function and operation are ignored.

Candera Objects in Toolbox to remove:

- Morphing Mesh, Stereo Camera, SkyBox, Cube Map, Point Sprite.

Render Targets:

- Remove Conformant Property (EGL Conformance)

Lights:

- 8 Lights are supported. Range property is ignored. All light types supported.

> Info - in simulation, Amber P renders objects without effects (default shader is associated) with OpenGL 1.1 API without shaders. SC Gizmos plus objects with special effects are rendered with OpenGL 2.0 via RenderDevice Extension.

SceneComposer Usability Improvements

Disable Widgets

In the "Preferences" menu, two new options for disabling widgets were added:

- Enable widget instantiation - true by default. If this option is disabled, the widgets will still be accessible in the solution, but they will not be instantiated by SCHost.
- Enable widget update - true by default, indicates whether or not the widgets will be updated.

See also:

- Widgets in SceneComposer User Manual

Convert Shaders

If any shaders or shader programs were modified or added in the new version of SceneComposer, the user will be notified during the conversion process. He will have to choose whether to replace them or not. In order to fulfil this operation in a suitable way, the user has the possibility to select those old shaders which has to be replaced with the new solution shaders and shader programs.

See also:

- Convert Solution in SceneComposer User Manual

2D Nodes: Object Inspector

Because of the difficulty to determine the absolute position or the effective size of a specific 2D node - especially when different layouters are used, an "Object Inspector" has been added. Due to this feature, now it is possible to display the following properties of a 2D node: Position, Scale, Rotation and BoundingBox.

WorldPivotPosition, WorldPosition and WorldBoundingBox can also be displayed in the Scene Editor panel. This feature is disabled by default and can be enabled in "Preferences" > "Editor" > "2D" > "World Transformations" (Mantis 5456).

See also:

- Position of 2D Nodes in SceneComposer User Manual

Preserve Imported Folder Structure

In the "Import" dialog window a new option was added: "Preserve imported folder structure". This option allows the user to import, simultaneously, many items without putting them together in the same folder.

> If a bunch of images are imported, any of them can be added in a folder instead of being added - all of them - together in the same folder.

It is available for the following imports: FBX, images, custom imports. This feature can be used only if the "Delete existing items in destination folder" is checked. Otherwise it will be unavailable and unchecked (Mantis 5444).

See also:

- Preserve Imported Folder Structure in SceneComposer User Manual

---

Disable Previews

In order to improve the performance of the application, now it is possible to enable/disable previews by checking/unchecking the "Disable Previews" checkbox from "File" > "Preferences" > "Environment" (Mantis 5509).

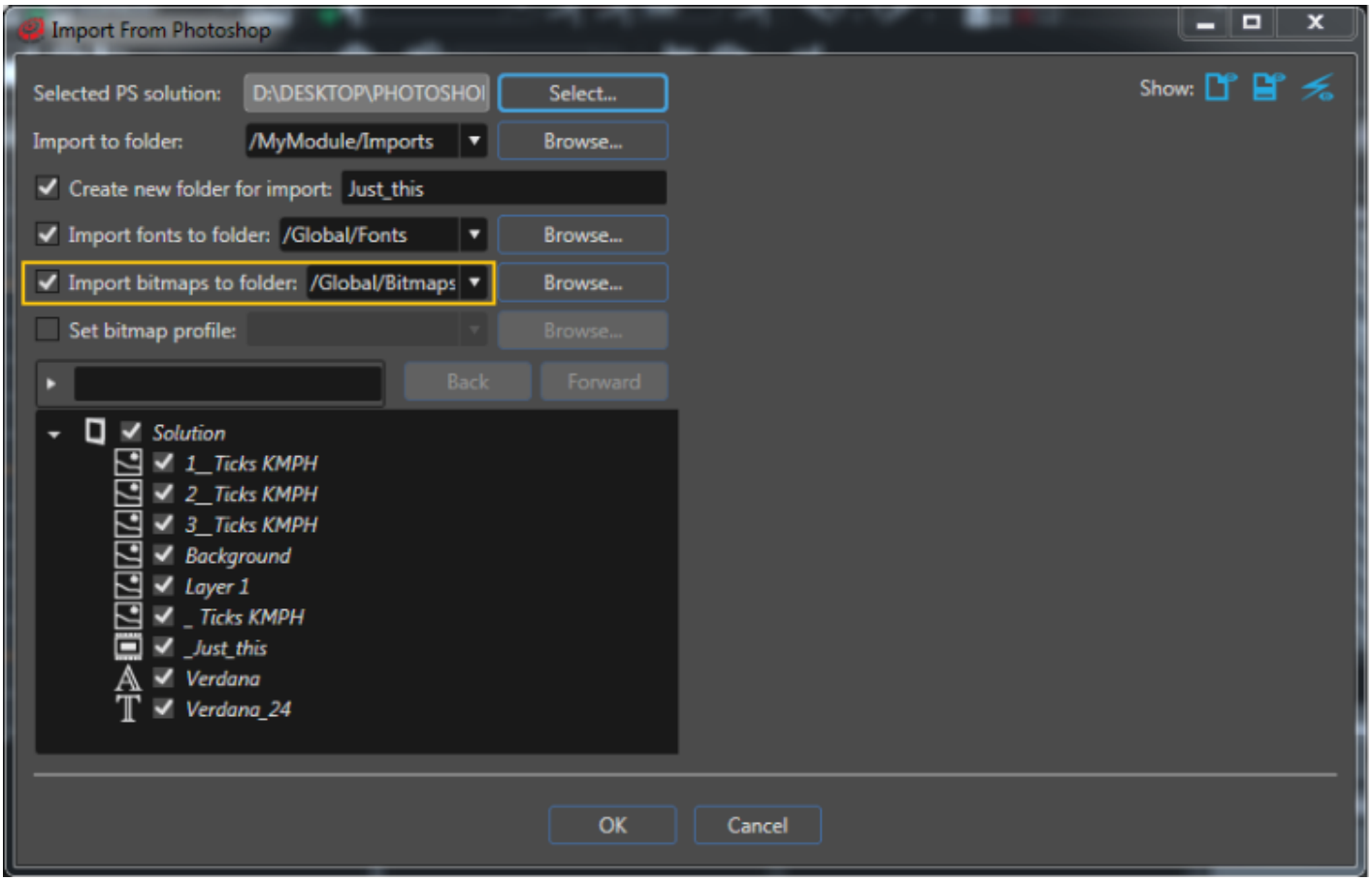
See also:

- Disable Previews in SceneComposer User Manual

Import Bitmaps in a Selected Folder

The user will have the possibility to import the bitmaps in a selected folder while using the "Import From Photoshop" dialog which is part of the "Import" menu located on the menu-bar.

To enable this option the checkbox which correspond to "Import bitmaps to folder" must be checked otherwise the bitmaps will be saved together with the other items in the import folder.



If this option is enabled the user will have the possibility to import the bitmaps in a selected folder. The selected folder will be saved after the dialog or the application will be closed and it will be available as predefined folder to import the bitmaps next time when the user will run again the "Import From Photoshop" dialog (Mantis 5484).

Asset Generation

It is possible now to exclude render targets from the asset. In the previous versions, all the render targets from all the displays were included in the asset. With this change, the asset will contain only those render targets which have the "Export always" property used or are referenced by scenes that are effectively included in the asset (Mantis 5306).

See also:

- Asset Library in SceneComposer User Manual

FTCCmd Support for Setting Bitmap Profile

It is now possible to specify a bitmap profile to be applied to all imported images when invoking FTCCmd.exe (by using parameter /bitmapprofile).

See also:

- FTC Command in SceneComposer User Manual

Minor Improvements

- Widget Properties. Until now, if a widget property was not available in SCHost.dll, SceneComposer would silently remove all values of this new property without any notification. Now, if this situation occurs, when opening a solution, the user is notified through a dialog that some of the widget properties were removed. He has the option to continue, thus losing the set properties at the first save, or to quit loading the solution (Mantis 5446).

- Background Color. It is now possible to set the background color for the Scene Editor camera in both 2D and 3D, both locally (for each scene) and globally, for all scenes (Mantis 4833).

- Mipmapping. "Enable mipmapping" property was moved from texture to bitmap. Solutions will have to be converted.

- Changing GUID. The solution GUID will be changed when changing the platform or converting to another version (Mantis 5452).
