CGI Studio 3.1.0

- Candera V3.1.0
  - Candera Engine 3D
  - Candera Engine 2D
  - Candera System
  - Candera Device
  - Candera AssetLoader
  - Candera Globalization
  - TextEngine
  - SCHost.dll
  - Migration Guide
- Courier V3.1.0
- FeatStd V3.1.0
- SceneComposer V3.1.0
  - Requirements
  - New Bitmap Formats
  - SceneComposer Usability Improvements
  - SceneComposer Known Issues

Candera V3.1.0

Release Date: July 23, 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_1_0.

Candera Engine 3D

Occlusion Culling

This new feature allows to disable rendering of the objects when they are not currently seen by the camera because they are obscured by other objects. Occlusion culling becomes enabled when the render strategy for camera is set to OcclusionCullingCameraRenderStrategy. During the rendering, all nodes initialized by the strategy will issue occlusion queries. If query results are available from the previous frame, they are evaluated and all nodes deemed occluded will have their bounding box rendered invisibly instead of the node itself. Visible nodes, or nodes not having a QueryProperty attached, will be rendered as usual. A new query to determine visibility for the next frame will be issued regardless the result of the previous query.

Drawcall Batching by implicit Geometry Instancing

On OpenGL ES3.0 platforms, Candera's renderer automatically batches draw calls of nodes where possible, and issues a single instanced draw call per batch (Geometry Instancing) instead of submitting an individual draw call per node.

To trigger draw call batching, the following prerequisites must be met:

- an instance-able shader must be used, and the corresponding Candera::Shader object must have its maximum instance count set accordingly.
- nodes must share the vertex buffer, shader, textures, and render mode
- nodes must not use multi-pass appearances
- nodes must not use object-space lighting

If prerequisites are not met, nodes will be rendered individually.

In order to achieve good batching rates, nodes should be sorted by shader, or grouped together using a separate camera to guarantee sequenced submission to the renderer.

Renderer Statistics

Renderer::GetStatistics() provides information on how many draw calls, instanced draw calls, instances, vertices, indices, uniforms and SetUniform() calls were submitted to the RenderDevice in the process of executing Renderer::RenderCamera() or Renderer::RenderAllCameras().

CameraRenderStrategy enhanced

CameraRenderStrategy has been enhanced with a new choice "SkipNode" for CameraRenderStrategy::RenderPassAction. The interface has been enhanced with CameraRenderStrategy::SetRenderPassCompleted() and CameraRenderStrategy::Render().

Candera Engine 2D

Animation Property Setter for SnapToDevicePixel

A new animation property setter has been added to animate the "SnapToDevicePixel" property.

See also:

- Animation::SnapToDevicePixelEnabled2DPropertySetter

Candera System

Localization

Interface for Internal::LocalizableStringData and Internal::ParameterizedLocalizableStringData has been changed.

Overloaded placement operators *new* and *delete* have been replaced by Internal::LocalizableStringData::Create and Internal::ParameterizedLocalizableStringData::Create methods.

These were introduced to better handle the allocation and creation of new instances. As the default placement *new* operator is used within the implementation of Create methods, the matching operator *delete(void* pMem, void*)* had to be added also.

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".

Candera AssetLoader

Maximum Bitmap Glyph Size

AssetDescriptor provides a new interface for obtaining the maximum size of the glyphs of all the fonts in the asset. The maximum size can be used to optimally initialize the decompression buffer of the TextEngine.

Initial Value of BitmapImage2D MemoryPool

AssetLoader automatically assigns the value BitmapImage2D::MemoryPool::FlashMemory during reading BitmapImage2D instances from asset if the asset repository is directly addressable (e.g. in Flash).

See also:

BitmapImage2D::SetMemoryPool()

Candera Globalization

Custom Localizer Support

CMake feature switch CANDERA_CUSTOM_LOCALIZER_ENABLED has been introduced to enable the use of a custom localizer. In case this flag is enabled (along with CANDERA_GLOBALIZATION_ENABLED), the DefaultLocalizer which loads LanguagePacks from assets will be disabled.

Furthermore the interface to set or get a Localizer in GlobalizationCultureManager class has been changed to SharedPointer instead of pointer.

Additional information on how to set up a custom localizer can be found here: Custom Localizer Support

TextEngine

Bitmap Font Sizes

Bitmap fonts have now an interface to acquire information on available sizes.

See also:

TextRendering::BitmapFont::BmFont::GetEffectiveFontSize()
TextRendering::BitmapFont::BmFont::GetFontSizeCount()
TextRendering::BitmapFont::BmFont::GetPossibleFontSize()

SCHost.dll

Source File Relocation

SCHost.dll source files **SCHost.h** and **dllmain.cpp** are no longer provided separately in *cgi_studio_courier* and *cgi_studio_dev*.
Instead, these files are now located in *cgi_studio_schost/src/SCHost*.

Migration Guide

Following an overview about interface changes between Candera V3.0.4 and Candera V3.1.0.

Description	Candera V3.0.4	Candera V3.1.0
Deprecated DynamicPropertyHost::BrowseableForDescendants, replaced with BrowsableForDescendants	`DynamicPropertyHost:`	`DynamicPropertyHost::Browsa`
Getter for Localizer changed to SharedPointer	`Globalization::GlobalizationCult` `().SetLocalizer(&Globalization::`	`Globalization::GlobalizationCultureMana` `().SetLocalizer(Globalization::Localize`

Setter for Localizer changed to SharedPointer	`Localizer* localizer = Globaliza`
	`if (localizer != 0) {`
	`...`
		`Localizer::SharedPointer localizer = Gl`
		`if (!localizer.PointsToNull()) {`
		`...`

Courier V3.1.0

3.1.0-1 Force Repaint in Simulation

A new method to force a repaint of the display content has been added, only for Simulation.

See also:

Courier::Renderer::ForceKickDisplay()

FeatStd V3.1.0

Release Info

- Release Date: July 23, 2015

No changes on top of version 3.0.4.

SceneComposer V3.1.0

Requirements

Requirements

For new ESSL 3.0 shaders, SceneComposer 3.1.0 requires a video card capable of compiling GLSL3.0 shaders (ESSL 3.0 shaders are available in newly created solutions based on OpenGL ES 3.0 Base solution).

New Bitmap Formats

New Bitmap Formats

SceneComposer can now import .EXR and .HDR files as RgbFloatingPointR11G11B10PixelFormat and RgbFloatingPoint9E5FPixelFormat respectively (Mantis 5504). New formats are also supported by .KTX extension. The following formats are now supported:

- Etc2CompressedRgbPixelFormat
- Etc2CompressedSrgbPixelFormat
- Etc2Alpha1CompressedRgbaPixelFormat
- Etc2Alpha1CompressedSrgbaPixelFormat
- Etc2EacCompressedRgbaPixelFormat
- Etc2EacCompressedSrgbaPixelFormat
- EacCompressedUnsignedRPixelFormat
- EacCompressedSignedRPixelFormat
- EacCompressedUnsignedRGPixelFormat
- EacCompressedSignedRGPixelFormat
- SrgbUnsignedBytePixelFormat
- SrgbaUnsignedBytePixelFormat
- RgbFloatingPointR11G11B10PixelFormat
- RgbFloatingPoint9E5FPixelFormat
- RgbaUnsignedIntRev10A2PixelFormat

The following conversions are possible between the new formats:

- Etc2CompressedRgbPixelFormat -> Etc2CompressedSrgbPixelFormat
- Etc2CompressedSrgbPixelFormat -> Etc2CompressedRgbPixelFormat
- Etc2Alpha1CompressedRgbaPixelFormat -> Etc2Alpha1CompressedSrgbaPixelFormat
- Etc2Alpha1CompressedSrgbaPixelFormat -> Etc2Alpha1CompressedRgbaPixelFormat
- Etc2EacCompressedRgbaPixelFormat -> Etc2EacCompressedSrgbaPixelFormat
- Etc2EacCompressedSrgbaPixelFormat -> Etc2EacCompressedRgbaPixelFormat

The following conversions are now possible for the classic formats:

- RgbUnsignedBytePixelFormat -> SrgbUnsignedBytePixelFormat
- RgbUnsignedBytePixelFormat -> SrgbaUnsignedBytePixelFormat
- RgbaUnsignedBytePixelFormat -> SrgbUnsignedBytePixelFormat
- RgbaUnsignedBytePixelFormat -> SrgbaUnsignedBytePixelFormat
- SrgbUnsignedBytePixelFormat -> RgbUnsignedBytePixelFormat

- SrgbUnsignedBytePixelFormat -> RgbaUnsignedBytePixelFormat
- SrgbaUnsignedBytePixelFormat -> RgbUnsignedBytePixelFormat
- SrgbaUnsignedBytePixelFormat -> RgbaUnsignedBytePixelFormat

See also:

- Color Depth Configuration and
- Bitmap Converter in SceneComposer User Manual
- Bitmap Profile in SceneComposer User Manual

SceneComposer Usability Improvements

Default Solutions

SceneComposer provides default solutions for both OpenGL ES 2.0 (for shading language 1.1) and OpenGL ES 3.0 (for shading language 3.0).

Multiple Clone Operation

It is possible to clone a node multiple times by using the "Multiple Clone" option. The Clone View was improved by adding a Multiple Clone tab that allows to specify the number of the clones (Mantis 4091).

See also:

- Clone, Delete, Rename Scene Nodes in SceneComposer User Manual

---

Planar Shadow Plane Alignment

A new property is available for planar shadows: AlignmentNode. This property exposes Candera's automatic alignment behavior. When the alignment node is set, Candera automatically aligns the shadow plane to the alignment node (which usually is the shadow receiver) (Mantis 5406).

See also:

- Planar Shadow in SceneComposer User Manual

Instanceable Shaders

New shaders were integrated into SceneComposer. They have "Inst" in their names and are special shaders used for instanced draw. When using an instance-able shader, uniforms that are declared as arrays in the shader, will not be exposed as an array in the SceneComposer view. Instead it has to be exposed as a single value.

When an instanceable shader is selected, the property grid will display the Max Instance Count value - represents the maximum number of element that the shader can render in a batch render (Mantis 5487).

See also:

- Instanceable Shaders in SceneComposer User Manual

Blend Operation Minimum and Maximum

Two new blend operations were introduced: Minimum and Maximum. It is possible to export Minimum and Maximum enumerators for the property RenderMode::BlendOperation (Mantis 5407).

Animating SnapToDevicePixel Property

SnapToDevicePixel property of 2D nodes can now be animated (Mantis 5689).

Tree Virtualization

Tree virtualization is now enabled on SceneComposer UI (several controls affected). Panel loading speed improved (Mantis 5666).

Occlusion Culling

A new camera render strategy was introduced: OcclusionCulling. When OcclusionCulling render strategy is selected, a new property becomes available in property grid: QueryType (Mantis 5488).

See also:

- Occlusion Culling in SceneComposer User Manual

Minor Improvements

- BitmapFont glyph size per Asset. CFF has been extended and a new information is available in the header of the asset files: maximum size (bytes) of bitmap font glyphs.

  This information is intended to be accessible to Candera asset loader and applications (Mantis 5517).
- Text Search in Shader Editor. Text search in shader editor was improved (Mantis 5500).
- Dragged Images on Billboards. Images can now also be dragged directly on billboards (Mantis 5623).
- Trigger Update. When a value of a property of a widget is changed, the Update method on the widget will be called immediately. This applies for composites too (Mantis 5668).
- Info Icon. A information icon for the default asset profile was added in order to inform the user that the changes (CustomID) will not be saved. (Mantis 5675).
- Bitmap Profile. "Bitmap Profiles" view was renamed into "Bitmap Profile Palette" (Mantis 5717).
- Additional Info. Additional information is displayed when adding a new TextStyle - "Base Style" and "Default Font" (Mantis 5777).
- Fixed angle rotation. The button is also displayed in "Animation Design" toolbar perspective. In "Scene Design" is always enable and the button position was changed near the "Rotation button" while in "Animation Design" is enabled only if an "Animated Property" is selected (Mantis 5625).
- In the SCHost Information panel new parameters were added (CGIDEVICE_OPENGLES_30 and CGIDEVICE_OPENGLES_20) in order to show which OpenGLES version (2.0 or 3.0) is used.

SceneComposer Known Issues

Due to enabling tree virtualization, some problems might appear:

- Vertical scroll bar changes its size randomly and doesn't remember sizes of elements after viewing the whole tree. Scrolling with mouse is hard.
- After some scrolling up and down, the application might freeze, especially in hierarchies with many levels.