Candera Engine 2D

Deterministic Render Order

In the past, RenderNodes with the same depth value have been rendered in an order which was dependent on the point of time the node has been added to the scene. The result was that in SceneComposer the render order could look different as when

loading the scene from the asset on target.

The new implementation ensures that the render order of nodes with the same depth value is defined by the position within the scene tree. Nodes which appear on top of the scene tree (as shown in SceneComposer) are rendered before nodes that are located on the bottom of the scene tree. Therefore the render order is always the same during design on host (SceneComposer) and on target.

Cloning Interface

In accordance to basic Unified Cloning Interface, the following changes have been applied to the affected classes in Candera Engine 2D:

- CloneSelf(CloneOperation) protected interface was removed.
- Clone(TraverseOperation, CloneOperation) has become deprecated.
- Copy constructor lost the optional CloneOperation argument.
- Clone() const function was added.

The cloning interface was applied to:

- Node2D and its descendants: Camera2D, CompositeGroup2D, Group2D, RenderNode, Scene2D.
- Effect2D and Layouter already define the interface appropriately.

Special considerations:

- General. Descendants of Node implement deprecated "Node2D* Clone(Traversing) const" and "Node2D Clone(TraverseOperation, CloneOperation) const" for backward

compatibility.

- Camera2D. Camera2DRenderStrategy and RenderTarget2D are not linked to the clone anymore.
- Node2D. Layouter is copied during deep node cloning. Because of this Node2D handles the destruction of Layouter upon disposing the node.
- Scene2D. Scene2D may be cloned. Scene2D render order is not cloned, as it doesn't support different render order bins.

Deep Cloning Interface

In accordance to basic Tree Cloning Strategies, the following support is provided for deep cloning in 2D as follows:

- TreeCloner2D is a specialization for the 2D scene tree of TreeClonerBase.
- DeepNode2DCloneStrategy provides a generic node strategy that clones the delegates the cloning of derived Nodes to specialized strategies.
- DeepCompositeGroup2DCloneStrategy resolves node binding by use of TreeMatch2D.
- DeepRenderNodeCloneStrategy resolves the sharing of effects by use of a map.
- DeepTreeCloner2D is a wrapper of DeepNode2DCloneStrategy that provides an interface similar to TreeCloner2D.

CompositeGroup2D

CompositeGroup is a new Node2D type that is usually created by AssetLoader and which contain:

- a list of AnchorPoints: descendants of the CompositeGroup2D that can be accessed directly from the CompositeGroup2D instance.
- a list of Widgets that act on any node within the CompositeGroup2D subtree.
- a list of AnimationPlayers that act on any node within the CompositeGroup2D subtree.

SharedPointer: Deprecated '...Ptr' types

All effect Ptr types like e.g. BitmapBrushPtr are marked as deprecated. Please use e.g. BitmapBrush::SharedPointer instead.

Deprecated BaseEffect2DPropertySetter

BaseEffect2DPropertySetter and its derivate classes became deprecated. Effect2D objects, as other objects with attached MetaInfo, can already animate their properties using built-in PropertyMetaInfo AnimationPropertySetters, with an improved performance.

AnimationPropertySetters Namespace Correction

All AnimationPropertySetters residing in `<cgi_studio_candera>/src/Candera/Engine2D/AnimationPropertySetters` have been moved from the *Candera* to the *Candera::Animation* namespace. The following classes are affected by this change:

- AlphaNode2DPropertySetter
- BaseNode2DPropertySetter
- BaseTransformable2DPropertySetter
- BaseTransformable2DRelativePropertySetter
- BaseEffect2DPropertySetter
- Effect2DFloatPropertySetter
- RenderingEnabledNode2DPropertySetter
- Transformable2DRelativeRotatePropertySetter
- Transformable2DRelativeScalePropertySetter
- Transformable2DRelativeScaleXPropertySetter
- Transformable2DRelativeScaleYPropertySetter
- Transformable2DRelativeTranslatePropertySetter
- Transformable2DRelativeTranslateXPropertySetter
- Transformable2DRelativeTranslateYPropertySetter
- Transformable2DRotatePropertySetter
- Transformable2DScalePropertySetter
- Transformable2DScaleXPropertySetter
- Transformable2DScaleYPropertySetter
- Transformable2DTranslatePropertySetter
- Transformable2DTranslateXPropertySetter
- Transformable2DTranslateYPropertySetter

> Moved classes are still accessible via the Candera namespace in CGI-Studio 3.0.0 but these deprecated interfaces will be removed in the next release.

Candera2D Bitmap MipMap Flag

Candera2D internally uses MipMapping for Bitmaps, as long as OpenGL is used to render. Whereas this measure typically improves image quality, MipMaps consume ~1/3 more memory and might bear side effects (interpolation between MipMap levels may produce artifacts). Therefore a MipMapEnabled flag has been added.

See also:

Candera::Bitmap::SetMipMappingEnabled() and Candera::Bitmap::IsMipMappingEnabled().

Camera2D Viewport Transformation functions moved

Viewport transformation functions have been moved from Candera::Camera2D to new class Candera::Math2D. Picking in 2D considers window offset.

Reduced size of members

To reduce heap usage some elements in Candera 2D have been reduced in size:

- The camera sequence number from 32 to 16 bit,
  See also:
  Candera::Camera2D::SetSequenceNumber(Int16 sequenceNumber)
- The bitmap width and height from 32 to 16 bit,
  See also:
  Candera::Bitmap
- The child count of Node2D from 32 to 16 bit,
  See also:
  Candera::Node2D::GetChildCount()

DeviceObject2D listener.

DeviceObject2D now supports adding listeners that are notified before and after the object is uploaded and unloaded and before it is destroyied.

Layout Rectangle

The Layout is no longer done based on bounding rectangle. Instead, Node2D has a new property, layout rectangle. Bouding rectangle is used for picking and dirty area management. Computed layout and bounding rectangle are based on affect layout and bounding rectangle, respectively.

See also:

> Node2D::SetLayoutingRectangle() and Candera::Effect2D::GetLayoutingRectangle().
>
> Candera::Node2D uses new Layout Rectangle instead of Bounding Rectangle for layout.

TextBrush and TextBrushCaches compute their Layout Rectangle based on Text Cursor Position.

MaximumSize property of TextBrush has been replaced by CacheArea. Bounding Box and Cache Size have been decoupled from Layout. To retrieve the height of a TextBrush with ConstantHeight use GetLayoutingRectangle, and for ActualHeight use GetBoundingRectangle.

Renderer2DListener

A Renderer2DListener allows to listen to Renderer-Events in Candera2D like in Candera 3D with RendererListener.

Revision #4
Created 2023-03-02 08:19:32 UTC by Kanai Tomoaki
Updated 2024-07-29 06:25:12 UTC by Tsuyoshi.Kato