Candera Related Features

Array Editor

Array properties are now supported for dynamic items (widget, effect, render target and layouter). Special editors are available in properties panel for changing the value.

See also:

- Candera::ArrayProperty

Candera Driven Validation

A new feature was added which enables Candera engine to validate the content from SceneComposer. For each dynamic property of widgets, effects, render targets or layouters, Candera engine can generate a problem message (error or warning). This message is displayed in the problems panel and also as a tooltip in the properties panel for the item on which the property is set.

If an item has properties with Error problems, the asset is not generated for it.

See also:

- Candera::DynamicProperties::ValidateValue

Camera Groups

The purpose of this work package is to support the creation of camera groups and to offer extended support for display visualization by enabling/disabling camera groups. In Candera, a CameraGroups is a class containing a list of Camera2D and Camera3D objects.

The camera groups were added and they allow the user to group together cameras from the entire solution. The camera groups are displays in a tab or the render targets explorer panel.

Widgets can be associated to camera groups (widget property of type CameraGroup* is supported). However, the widget should not modify the state of the camera group (shall not add or remove cameras)

See also:

- Camera Group in SceneComposer User Manual
- Candera::CameraGroup

Composite Animations

The Composite Animation was Removed. An animation can now contain animated properties for composites and scenes (they can also be mixed).

See also:

- Add Animated Property in SceneComposer User Manual

Clear Mode

A new item of type "Clear Mode" can be created in SceneComposer and referenced from any render target.

See also:

- Clear Mode Item in SceneComposer User Manual
- Candera::ClearMode

Items AssetId

All items in a solution have a type and a name which identifies them in their parent container and a full name which uniquely identifies them in the entire solution (for example a mesh in a scene could have the full name /Global/Scenes/MyScene/Group:RootNode/Mesh:MyMesh).

Similar to the full name which identifies the items in the solution, for Candera there is a CanderaPath which uniquely identifies an item in an asset file (for example the mesh mentioned above will be referenced in Candera using the path /Scene:Global::Scenes::MyScene/Group:RootNode/Mesh:MyMesh). Such a string will have to be stored in the asset library for animations or widgets which reference scene nodes. Also it will be processed every time the scene is loaded.

> Candera does not have the concept of solution folders, so in order to distinguish between items with the same name but located in different solution folders the CanderaName

> property of the items contain also the name of the folders (for example the scene /Global/Scenes/MyScene will have the CanderaName Global::Scenes::MyScene).

To improve the performance we introduced another way to identify items using numerical Ids instead of strings.

- **Symbolic Names.** This type of names can be assigned to items to make their Id accessible from applications. A C++ header with all the symbolic names from the solution can be generated via File->Export Symbolic Names.

See also:

- Items AssetId in SceneComposer User Manual

EGL Context Sharing

A Context Resource Pool (CRP) is used to share resources among different Context Providers. It allows grouping of render targets and associated device objects (Shaders, Texture Images, Vertex Buffers, etc.) in a common pool.

Each platform has some rules and restrictions related to CRPs, for example on some platform multiple displays can share the same CRP, while on some other platform a CRP is required for each Window Surface. SceneComposer displays the relationship between CRPs, Displays, Window Surfaces and Frame Buffer Objects in the Render Target Explorer panel (main view and an additional CRP view). Depending on the platform, the texture render targets (frame buffer objects) may have an Owner property of type render target reference or CRP.

If the type of the Owner property is render target reference, the owner must be a display render target (window surface).

If the type of the Owner property is CRP (integer value in SceneComposer), the possible values will be obtained from the available displays.

If the Owner property is available for a render target but it is not set by the user, the render target will be considered orphan.

See also:

- Context Resource Pool in SceneComposer User Manual
- Candera::ContextResourcePool

BitmapFormat and BitmapType

These properties were made obsolete and their value were merged into PixelFormat which contains only the valid combinations of the former properties.

Revision #3
Created 2023-03-02 09:33:20 UTC by Kanai Tomoaki
Updated 2023-06-13 22:50:12 UTC by Kanai Tomoaki