Behavior Definition

Behavior Definition

While SceneComposer can be used to compose the static structure and properties of a scene, with Behaviors it is possible to add dynamics to a static scene structure.

- A Behavior is defined by behavior attributes: Name, Type, Properties
- Behavior attributes are expressed in the Behavior Meta Information
- Behaviors are assembled together in a widget set
- Behaviors and the widget set can be assembled in several Behavior libraries (lib), to be linked to the application

Behavior Generation

If *cgi_studio_cit* is within the package, the base classes for Behaviors can be generated. Therefore an .xhcdl-file has to be written. The base class can include properties that can be defined in the .xhcdl-file too. The xhcdl-files can be found for each behavior in the same folder where the source-code lies. The following example shows an .xhcdl-file for a behavior.

```
<?xml version="1.0" encoding="utf-8"?>
<definition
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="../../../../cgi_studio_cit/schema/HmiContract.xsd">
    <include href="Behaviors/Behavior.xhcdl" addIncludeToCode="false" />
    <generator location="Behaviors/Transition/generated">
        <namespace namespace="Candera" />
        <header>
            <include file="Behaviors/ActionBehavior/ActionBehavior.h" />
            <include file="CanderaWidget/WidgetBase/StringDataTypes.h" />
        </header>
    </generator>
    <widgets setPolicy="onChange" notifierPolicy="onUpdate" viewInvalidationPolicy="wakeup">
        <group baseClassRef="ActionBehavior">
            <widget name="SimpleTransitionActionBehavior" readableName="" uid="{889EF594-E702-48
                <property name="ActivatedSceneAssetID" description="The scene to activate." type
                <property name="IsActivatedScene3D" description="If the activated scene is 3D or
                <property name="DeactivatedSceneAssetID" description="The scene to deactivate."
                <property name="IsDeactivatedScene3D" description="If the deactivated scene is 3
                <property name="Variant" description="The transition variant." type="FeatStd::St
            </widget>
        </group>
    </widgets>
```

```
</definition>
```

Some important tags and attributes:

- Generator location: Defines in which folder the base class should be generated.
- Group baseClassRef: The base class of the generated base class.
- Widget name: The name of the generated base class. The suffix "Base" will be added to the class-name.
- Property: The properties of the Behavior. For the String-type the include file *CanderaWidget/WidgetBase/StringDataTypes.h* has to be added for instance. If the property can't be generated it can still be added in the source code.

After the .xhcdl-file is written the behavior can be generated by calling *GenerateBehaviors.bat* which can be found in *cgi_studio_controls/src/Behaviors*. The files are a base class with a header- and cpp-file. It is meant to derive from that class. All the properties and its getter- and setter-methods are already in the class.

Behavior Meta Information

Behavior Name and Type

To publish a behavior to SceneComposer, it must register for the widget with its name and type:

CdaBehaviorDef	Behavior class name and its base class
CdaDescription	Behavior description for the SceneComposer tooltip
CdaReadableName	Behavior name

> Note that the example above explains how to publish a behavior for a 3D-scene. For 2D-scenes the macro *CdaBehavior2DDef* must be used instead of *CdaBehaviorDef*.

Behavior Properties

For linking a behavior with scene content in SceneComposer, behavior properties are used. Usually the properties should be defined in the .xhcdl-file of a behavior. Anyway, the properties can be defined directly in the behavior-class by using the appropriate macros in the header-file. The behavior implements a property by providing a getter and a setter method.

CdaProperty	Specifies [property_name], [property_type], [property_getter], [property_setter]
CdaDescription	Behavior property description for the SceneComposer tooltip

CdaCategory	Defines a category for the property. SceneComposer displays all properties of one category in a related Category section in the property grid.

> The Behavior property must be initialized correctly, e.g. in the widget constructor!
> Take care that the getter and setter methods are safe (prevent null pointer accesses)!

Usage of the introduced macro CGI_BEHAVIOR_FORWARD_AssetLoaderDataTypeDef

This macro is used to be able to set a certain behavior as property of another behavior.

Therefore following code has to be added to the xhcdl file to generate the macro in the base class of the behavior:

```
<postNamespace text="CGI_BEHAVIOR_FORWARD_AssetLoaderDataTypeDef(::CgiStudioControl::ValueBehavi
```

This is an example code to be able to use a ValueBehavior as property. The generated code in the base class of your behavior will look like this:

```
CGI_BEHAVIOR_FORWARD_AssetLoaderDataTypeDef(::CgiStudioControl::ValueBehavior)
```

Revision #3
Created 2023-02-24 01:38:31 UTC by Kanai Tomoaki
Updated 2023-06-19 02:26:40 UTC by Tsuyoshi.Kato