

Configuring and Loading an Animation

Candera Animation Concept

The following key concepts are required to run animations in a [Candera](#) application:

- [Candera::Animation::AnimationPlayer](#) provides the control interfaces (use [Candera::Animation::AnimationGroupPlayer](#) for animation groups)
- [Candera::Animation::AnimationTimeDispatcher](#) dispatches the current world time to animations; animations can also query the current world time from the dispatcher
- [Candera::Animation::AnimationKeyframeSequence](#) configures proper animation values for each time point of an animation.
- [Candera::Animation::InterpolationStrategy](#) specifies for a key frame sequence, how values between several key frames shall be calculated.
- [Candera::Animation::AnimationPropertySetter](#) applies animation values to animated properties

In SceneComposer, animations can be configured in the AnimationDesign view:

- Create a new animation - [New Animation](#) in SceneComposer User Manual
- Select a node property to animate - [Add Animated Property](#) in SceneComposer User Manual
- Specify the key frame sequence including property values and interpolation strategy - [Animation Values](#) in SceneComposer User Manual

It is also possible to import an animation to SceneComposer from a content generation tool via FBX format - [How to Import Resources](#) in SceneComposer User Manual

Configuring Animation Player

In the Player or in an application, the animation can be retrieved from the generated asset by specifying its [Candera](#) name (e.g. *MyModule#Animations#Animation*).

```
AssetProvider* assetProvider = GetAssetProvider();  
m_animPlayer = assetProvider->GetAnimation(m_animName);
```

In SceneComposer the animation can be retrieved from AssetProvider only if it was previously loaded in the scene. A better way to access it directly is to define it as a widget property.

```
// Define Animation property
CdaProperty(Animation,Candera::Animation::AnimationPlayer::SharedPointer, GetAnimation)
    CdaDescription("Animation")
CdaPropertyEnd()
// End Define Animation property
```

Next, as already explained in [Initializing AssetLoader and Animations](#), an instance of [Candera::Animation::AnimationTimeDispatcher](#) is needed to provide current world time to animations.

For this, the new animation player must be added to the [Candera::Animation::AnimationTimeDispatcher](#) instance maintained by the application.

```
// Add player to time dispatcher

Animation::AnimationTimeDispatcher::SharedPointer timeDispatcher = GetAnimationTimeDispatcher();
if (timeDispatcher == 0) {
    return;
}
static_cast<void>(timeDispatcher->AddPlayer(m_animPlayer));

// End Add player to time dispatcher
```

Now the animation player is ready to be used.

Revision #10

Created 2023-02-22 01:48:52 UTC by Tsuyoshi.Kato

Updated 2024-11-07 01:14:37 UTC by Tsuyoshi.Kato