Candera 3.6

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_6_0.

- Candera/FeatStd CMake Build System Changes
- Candera Engine 2D
- Candera Engine 3D
- Transitions
- Textengine

Candera/FeatStd CMake Build System Changes

Added flags:

Name	Description	Possible values
CANDERA_TEXTENGINE_NATIVE_LINE HEIGHT_ENABLED	Enables the native line height metrics of the font engine. If disabled then 120% of EM size is applied as line height which is compatible with Adobe.	ON/OFF

Candera Engine 2D

Node2d GetComputedBoundingRectangle, IsInsideBoundingRectangle, IsPickIntersectingBoundingRectangle and GetEffectiveBoundingRectangle interface extended

The method 'virtual void Node2d::GetComputedBoundingRectangle(Rectangle& boundingRectangle, bool ignoreInvisible = false) const' has been extended by the additional parameter ignoreInvisible. The default value of this parameter is false for full usage backwards compatibility. If the parameter is set to true only visible parts of the scene will be considered. This also means that clipped areas are not considered for picking and/or bounding box calculations. As a consequence it is now possible to pick only visible parts of the scene.

Pivot point for Transformable2D

New methods to Transformable2D have been added that allow to manipulate the real pivot point.

Those are Transformable2D::SetPivotPoint and Transformable2D::GetPivotPoint .

Candera Engine 3D

Generic Transform is Optional

The Property "Generic Transform" of a 3D Node became optional. The reason for that is to have as few matrix multipilcations as possible which saves performance. Further wise Generic Transforms are only used for special needs.

Pivot point for Transformable

New methods to Transformable have been added that allow to manipulate the real pivot point.

Those are Transformable::SetPivotPoint and Transformable::GetPivotPoint .

Transitions

Sequential Transitions - Extended Hint Constructor

The constructor for Transition Hints has been extended in order to reflect the newly added ability to execute Transitions in a sequential manner.

The added parameters are:

```
const FeatStd::Optional<FragmentStrategy>& activationStrategy
const FeatStd::Optional<FragmentStrategy>& deactivationStrategy
const FeatStd::Optional<Float>& activationDelay
const FeatStd::Optional<Float>& deactivationDelay
```

FragmentStrategy is a new enum type consisting of:

- Normal
- Early
- Late

Transition Rules - Check if Rule is Bidirectional

The Rule class has a new method called bool Rule::IsBidirectional() const, which returns, whether the Transition Rule is set to be bidirectional or not.

Textengine

AssetFontStore support for Freetype streams

AssetFontStore is extended to support two font load strategies:

- AssetFontStore::LoadStrategySelector::PreloadStrategy (the old behavior before the extension): entire fonts are loaded into memory once and font data is accessed from there by Text Engine.
- AssetFontStore::LoadStrategySelector::OnRequestLoadStrategy (new behavior, selectable through a strategy selector): small font portions are streamed directly from the asset repository when Freetype (or indirectly Harfbuzz) needs them.

Use DefaultLoadStrategySelector::SetDefaultFontLoadStrategy to select one of these load strategies. Use DefaultLoadStrategySelector::SetDefaultFontLoadStrategyExceptionList to exclude a list of fonts from the selected font load strategy.

Example:

```
    // Load fonts with OnRequestLoadStrategy
    selector.SetDefaultFontLoadStrategy(
Candera::AssetFontStore::LoadStrategySelector::OnRequestLoadStrategy);
    // "Bitstream Vera Sans" font shall be loaded with PreloadStrategy:
    const Char* g_fontName[2] = { "ConstructionKit##ConstructionKit#Resources#Fonts#Bitstream Ve
    selector.SetDefaultFontLoadStrategyExceptionList(g_fontName);
    fontStore.SetLoadStrategySelector(&selector);
```

Courier applications can use Courier::ViewHandler::SetFontStoreProviderCallback to provide specific settings for the Candera::AssetFontStore in use.

Support of Monotype's WorldType-Shaper

Candera supports the WorldType-Shaper of Monotype now. The integration is done to support the font engine 'iType'. With this release complex texts can be shaped. The behavior is similar to HarfBuzz with the difference of a huge performance boost. WT-Shaper may have a higher memory consumtion as a trade-off to performance. The implementation concept is aligned with HarfBuzz to support a flexible exchangeability.

With this the default shaper of iType switches from ComplexScript to WT-Shaper. The corresponding CMake flag CANDERA_TEXTSHAPER has been extended by the entry WorldType-Shaper.