Structural View

Design Constraints

Layouters for 2D scenes

CGI Studio Layouters are currently available for the 2D scene environment only. However the application of layouters with the 3D Canvas is being prepared.

Disable 2D layout

To save computing power and code size for low end devices, the CMake feature flag CANDERA_LAYOUT_ENABLED has been introduced. This flag allows to remove all code needed for dynamic layout calculation. To achieve this, the complete code for layout calculation are pooled in following folder:

cgi_studio_candera/src/Candera/Engine2D/Layout

By default, layout calculation is enabled, the flag is set to ON.

Design Considerations

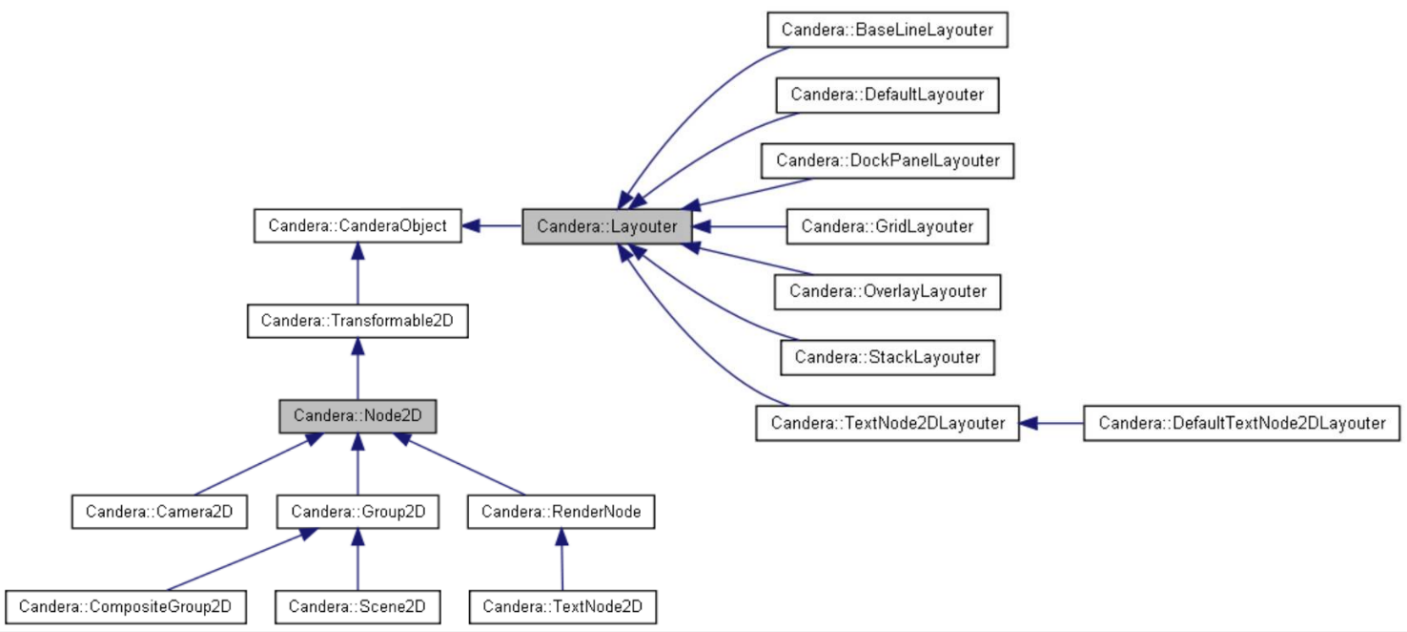
Default Layouter

For all nodes there is a Default Layouter available that behaves conform to the WPF Canvas control: Elements can be put into the scene and positioned using explicit coordinates.

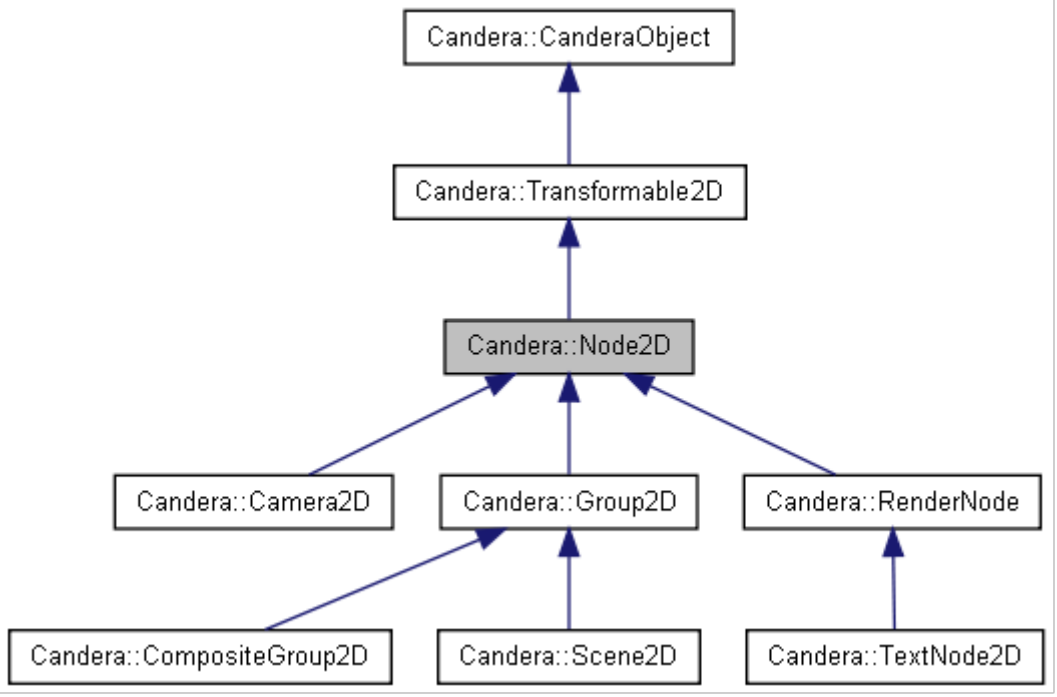
Dynamic Properties

Layout properties have been realized as dynamical properties of nodes. This has been done due to the fact that the use of layouters is optional and this way, memory can be saved.

Software Elements



Node2D and its Subclasses



Node2D
The class Node2D is an abstract base class for all 2D scene graph nodes. Each node defines a local coordinate system relative to the coordinate system of the parent node. The functionality to define the local coordinate system is derived from Transformable2D and consists of following components: translation, rotation, scale and a generic matrix called transform matrix. If the node is transformed from the local coordinate system to the world's coordinate system, then the node's local transformation is multiplied with all its parents’ transformations. A Node2D can also store a set of nodes as its children but a node can at most have one parent at a time. Cycles are prohibited.

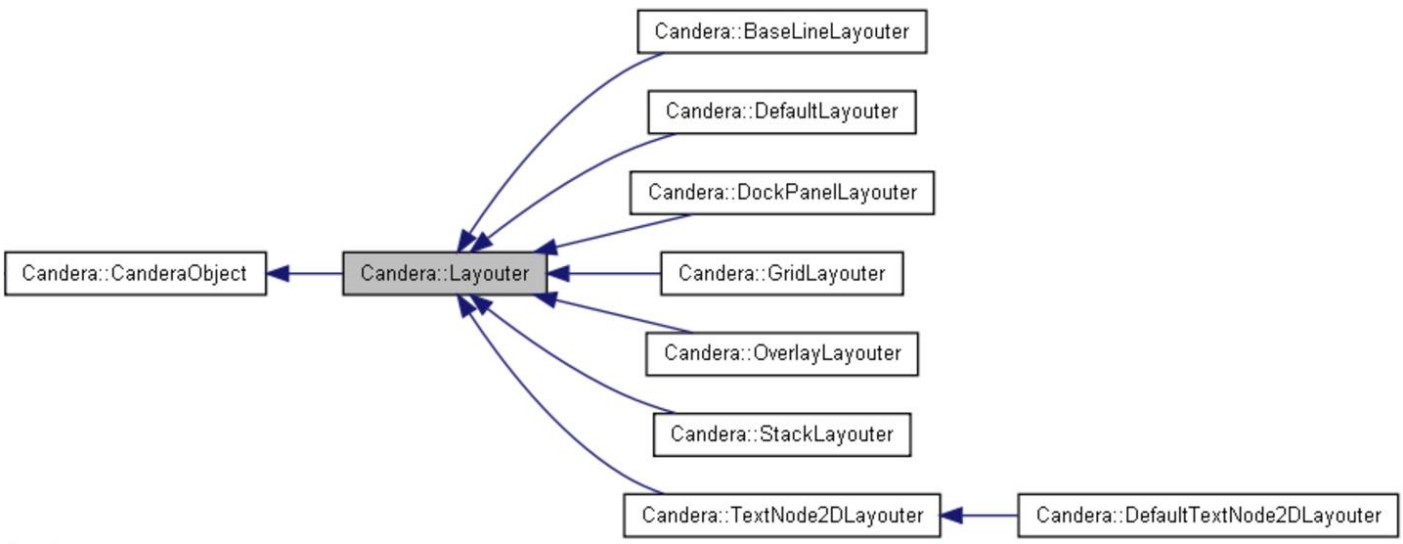
RenderNode
RenderNode is a 2D node that links to an effect chain. Therefore this is the only type of 2D node which has content to be rendered. All nodes of type RenderNode are added to a list which is sorted regarding each node's render order rank. The render order rank can be set using the SetRenderOrderRank method. The render order rank defines the relative value within render order. Nodes with lower render order rank are rendered before nodes with higher value, and therefore appear in the background. The effective render order rank is the sum of this node's and all its ancestor nodes' render order rank values.

TextNode2D
TextNode2D is a RenderNode that renders text using the attached BitmapBrush effect. To render text a FeatStd::String and a TextRendering::Style needs to be provided, as well as a TextNodeRenderer object which implements the strategy for providing Image2D objects, which will be rendered by the associated BitmapBrush. For text layout a TextNode2DLayouter should be attached to the TextNode2D.

Group2D
The class Group2D is a scene graph node that stores a set of nodes as its children. The parent-child relationship between groups and nodes is bidirectional. Whereas a group can hold a group of nodes, a node can at most have one parent at a time. Cycles are prohibited. Changing the groups' transformation or alpha value affects the child nodes of the group.

Scene2D
The class Scene2D represents a top level scene graph node. Multiple scenes are allowed. A scene cannot be part of any other scene.

Layouter and its Subclasses



Layouter
To layout a group of elements an instance of a Layouter has to be attached to a node. Child nodes act automatically as elements for the parent layouter. The order of the child nodes defines the order of the layout items.

DefaultLayouter
DefaultLayouter is one static instance used by all 2D nodes by default. This allows all nodes to get laid out without the need of adding a specialized layouter.

DockPanelLayouter
The Dock PanelLayouter provides an easy docking of elements to the left, right, top, bottom or center of the panel. The dock side of an element is defined by the attached dynamic DockSide property. To dock an element to the center of the panel, it must be the last child of the panel.

BaselineLayouter
A BaselineLayouter behaves similar to a Horizontal StackLayouter but instead of aligning the objects to the client area of the Layouter, it aligns them to a line. Horizontal stretching is disabled to avoid greedy items. Vertical stretching is disabled as result of DefaultLayouter::GetScale. The Vertical Alignment is interpreted as follows:

- VTop: the top of the child area is aligned to the base line.
- VBottom: the bottom of the child area is aligned to the base line.
- VCenter: the point defined by the child as its base point is aligned to the base line. This is typically in the center of an image, or on the base line of a text.
- VStretch: same as VCenter, as stretching is disabled. The baseline can be either fixed or automatic. The automatic baseline is configured such that all the children fit from the top of the client area.

TextNode2DLayouter
Base Layouter for TextNode2D nodes. The TextNode2DLayouter arranges and truncates the text associated with a TextNode2D.

DefaultTextNode2DLayouter
Default layouter for TextNode. The DefaultTextNode2DLayouter arranges and truncates the text associated with a TextNode2D.

StackLayouter
A StackLayouter stacks its elements either in horizontal or vertical order.

- Vertical stack: The preferred width of the stack is defined by the widest element.
- Horizontal stack: The preferred height of the stack is defined by the highest element.

OverlayLayouter
The OverlayLayouter sets sizes of all child elements to fulfill the given area.

GridLayouter
The GridLayouter arranges its elements in a grid. The amount of rows and columns for the grid must be specified. Each row and column can be configured with one float value:

- Automatic size (value < 0): The row height (or column width) will be defined by the highest value of its child elements. So the row or column is given exactly the amount of

space it needs, and no more.
- Proportional size (0.0 < value < 1.0): The space is divided between a group of rows or columns.
- Absolute size (value >= 1.0): Defines an exact (fixed) size in pixels.

Revision #3
Created 2023-02-27 08:01:05 UTC by Tsuyoshi.Kato
Updated 2023-06-13 05:18:14 UTC by Kanai Tomoaki