

Overview

General demands for Layouters

In a typical user interface the users need to distribute screen space to different elements. If the size of the elements is dynamic, it's desired that the screen space distribution is done automatically based on the content's natural size. To specify the behavior of this automatic layout, different layouters are used, each of which has its own layout strategy.

Adaption to the Window or the Display

Fixed pixel arrangement of controls doesn't work when the application needs to run on different targets. When creating a UI, the GUI elements (CGI-Studio nodes) are arranged by location and size to form a layout. A key requirement of any layout is to adapt to changes in window size and display settings. CGI-Studio layouters provide the possibilities to adapt a layout in these circumstances. Relative positioning is a keystone of the CGI-Studio layout system. It increases the ability to adapt to changing window and display conditions. Additionally, the layout system manages the communication between different elements to determine the layout: In a first step, a component tells its parent what size at what position it needs to have and in the second step, the parent tells the component what space it may have. In short, laying out is a recursive system that leads to an element being sized, positioned, and drawn.

Choosing a Layouter

The laying out of controls is very important and critical for an application's usability. It is used to arrange GUI elements in an application. The following important things have to be considered when selecting a layouter:

- Positions of the child elements
- Size of the child elements
- Layering of overlapping child elements on top of each other

Layouter Types

The CGI-Studio layout system provides several layouters for various use cases:

The **Default Layouter** is used to position elements explicitly using absolute coordinates. When no layouter is selected, the default layouter is used.

With the **Grid Layouter** child elements can be positioned in a tabular structure of rows and columns.

The **Overlay Layouter** can be used to arrange elements one over another to end up having e.g. a foreground element and a background element.

The **Stack Layouter** simply stacks the elements together under consideration of their preferred size and its configured horizontal or vertical alignment.

The **BaseLine Layouter** works similar to the Horizontal Stack Layouter, but the child elements are aligned to a line instead of aligning them to the client area.

The **DockPanel Layouter** allows you to snap, or dock, elements to the edges (top, bottom, left or right) of the rectangular client area. The docking edge can be defined for each child element. One child element is docked after the other, each leaving the remaining space for the rest of the elements, considering the order in which the child elements are added.

Revision #1

Created 2023-02-27 07:58:31 UTC by Tsuyoshi.Kato

Updated 2023-02-27 07:58:50 UTC by Tsuyoshi.Kato