
Specification proposal for resizing replacement in R3 GUI framework

Author: Richard Smolak

Revision History
Version
0.7

ARS

Table of Contents

1. Purpose	1
2. Overview	1
3. Specification	1
3.1. Resizing algorithms	1
3.2. Box model	3
3.3. GOB additional attributes	4
3.4. Attribute formats and units	4

1. Purpose

The purpose of this document is to design specification proposal for R3 GUI resizing replacement subsystem.

2. Overview

Current Resizing in R3 GUI has bugs and limitations. We need better resizing with following features:

- flexible resizing algorithm
- built-in box model
- pixel, percent, decimal units support
- easy to use by layout makers
- easy to understand by style makers

3. Specification

3.1. Resizing algorithms

To achieve great flexibility in the resizing of GOBs two possible algorithms are offered by the system which can be used and combined in different ways:

RESIZE mode

Overview

The RESIZE mode is useful for GOBs which offsets and/or sizes should copy exactly the amount of changed pixels during the resize event.

Algorithm

If RESIZE mode is enabled the offset and/or size of GOB is increased or decreased *absolutely* by the difference(delta) of the the desired and initial size of the root GOB.

Simplified algorithm looks like this:

```
delta-size = desired-size - root-gob/init-size
then:
gob/offset = delta-size + gob/init-offset
and
gob/size = delta-size + gob/init-size
```

Example

A good example is GOB used as a window background. If you enable the RESIZE mode for width and height of such background GOB and set its size to 100% of the parent GOB(window) size it will fill the whole window area exactly according to the window size.

RESCALE mode

Overview

The RESCALE mode changes the offset and/or size of GOB proportionally so the initial size/offset ratios are achieved. Simply said the GOB is *rescaled*.

Algorithm

If RESCALE mode is used the offset and/or size of GOB is increased or decreased *proportionally* by multiplication of the percentage ratio of desired and initial size of the root GOB.

Simplified algorithm looks like this:

```
ratio = desired-size / root-gob/init-size
then:
gob/offset = ratio * gob/init-offset
and
gob/size = ratio * gob/init-size
```

Example

The RESCALE mode can be used for example for TABLE row layout where we need that the cell widths should keep their ratio and scale proportionally.

Note

The examples of usage of different resizing modes mentioned above are just for illustrational purposes. It is possible to combine the modes in many ways and even separately for GOB sizes or offsets to achieve specific results.

3.2. Box model

The used box model is in some ways similar to well known CSS box model type. Every GOB has a content area and optional surrounding padding, border, and margin areas.

The margin, border, and padding can be broken down into top, right, bottom, and left segments.

The perimeter of each of the four areas (content, padding, border, and margin) is called an "edge", so each box has four edges:

- content edge*

The content edge surrounds the content area of GOB. The four content edges define the GOB's content box.

- padding edge*

The padding edge surrounds the GOB padding. If the padding has 0 width, the padding edge is the same as the content edge. The four padding edges define the GOB's padding box.

- border edge*

The border edge surrounds the GOB border. If the border has 0 width, the border edge is the same as the padding edge. The four border edges define the GOB's border box.

- margin edge*

The margin edge surrounds the GOB margin. If the margin has 0 width, the margin edge is the same as the border edge. The four margin edges define the GOB's margin box.

Note

The margin area has transparent background so it reveals the content under the GOB. While the background of padding area is of the same color as background color of the GOB.

Each edge may be broken down into a top, right, bottom, and left edge. The offset/size values for each edge have to be specified in pixel units.

- GOB content area size*

The dimensions of the content area of a GOB can be affected by the settings of all edges so the final width and height of the GOB is always equal to the defined size of the gob.

So the following equation is always true:

$$\text{content-size} = \text{gob-size} - \text{margin-top-left} - \text{border-top-left} - \text{padding-top-left} - \text{margin-bottom-right} - \text{border-bottom-right} - \text{padding-bottom-right}$$

For example GOB of size 100x100 pixels with border of 1 pixel on all edges will have content area of size 98x98 pixels.

3.3. GOB additional attributes

The resizing system needs to define few additional attributes for each GOB to be able work properly.

init-offset	holds initial offset of the GOB (default is 0x0)
init-size	holds initial size of the GOB (default is 100x100)
min-offset	defines minimum GOB offset value boundary that will be used during calculations (default is none!)
min-size	defines minimum GOB size value boundary that will be used during calculations (default is none!)
max-offset	defines maximum GOB offset value boundary that will be used during calculations (default is none!)
max-size	defines maximum GOB size value boundary that will be used during calculations (default is none!)
resize-xy	enables RESIZE mode for GOB offset (default is none!)
resize-wh	enables RESIZE mode for GOB size (default is none!)
rescale-xy	enables RESCALE mode for GOB offset (default is none!)
rescale-wh	enables RESCALE mode for GOB size (default is none!)
border-color	defines color of GOB border (default is 'black')
border-size	defines size(s) of GOB border (default is 0)
margin	defines margin offset(s) of GOB (default is 0)
padding	defines padding offset(s) of GOB (default is 0)
draw	holds user defined graphical content of GOB in form of DRAW dialect block
size	holds GOB content size. (reference value that can be used inside DRAW dialect)
space	holds summed top-left/bottom-right space values between GOB content and margin (used internally for box model calculations)

3.4. Attribute formats and units

This section will describe data formats used by specific attributes.

INIT-OFFSET, INIT-SIZE, SIZE

format			description	
pair!			offset/size as pair	

Specification proposal for resizing
replacement in R3 GUI framework

[number! number!]			offset/size as integer or decimal form	
-------------------	--	--	--	--

MIN-OFFSET, MIN-SIZE, MAX-OFFSET, MAX-SIZE

format			description	
--------	--	--	-------------	--

RESIZE-XY, RESIZE-WH, RESCALE-XY, RESCALE-WH

format			description	
--------	--	--	-------------	--

BORDER-COLOR

format			description	
--------	--	--	-------------	--

BORDER-SIZE, MARGIN, PADDING

format			description	
integer!			all edges will have the same integer value	
pair!			top/bottom and left/right edge pairs will have the same value	
[pair!]			only the top/left edges are set	

DRAW

format			description	
--------	--	--	-------------	--

SPACE

format			description	
--------	--	--	-------------	--