

Pyramidion : a framework for Domain-Specific Editor

IWST 2024

Yann LE GOFF yann.le-goff@thalesgroup.com

Pierre LABORDE pierre.laborde@thalesgroup.com

Alain PLANTEC alain.plantec@univ-brest.fr

Eric LE PORS eric.lepors@thalesgroup.com

THALES
Building a future we can all trust

UBO
Université de Bretagne Occidentale

Lab-STICC
UMR 6285

www.thalesgroup.com

www.univ-brest.fr

www.labsticc.fr

Contents

03



Definition

04



Example

06



Architecture

18



Validation

Why do you need a Domain-Specific Editor ?

> Create a Graphical User Interface

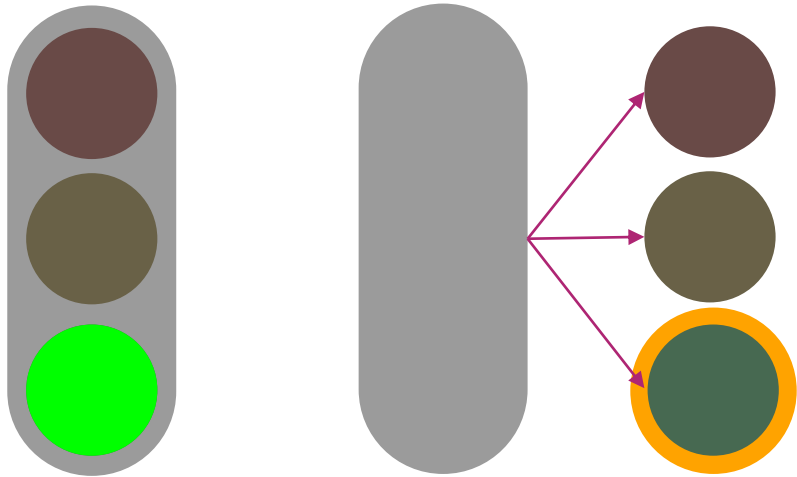
- ▶ Choose a GUI library and a GUI language
- ▶ Writing Code
 - It is slow
 - Execute the code to know what you are doing
 - You need to either know the language or refer to the documentation
- ▶ Using a GUI editor
 - It is fast
 - WYSIWYG
 - The possible modifications are displayed in the dedicated tool
 - The possible modifications are limited by the tool

> Pyramidion

- ▶ A framework to create Domain-Specific editor
- ▶ Used to create Pyramid a GUI Editor.

Domain = objects modified by the editor

A Traffic Light example (1/3)



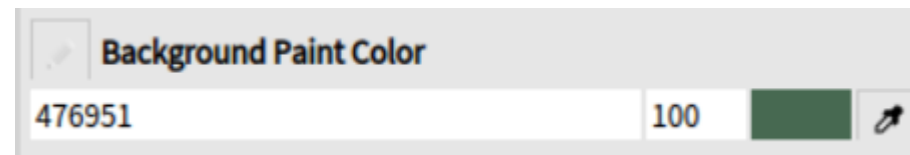
> Step 1 : Select the element

- ▶ I need a widget to select the element
- ▶ I need a selection system

Elements	Hash	z	Visibility
▼ light_frame <i>BEElement</i>	314609408		👁 visible
○ red_light <i>BEElement</i>	100126694		👁 visible
○ yellow_light <i>BEElement</i>	425302528		👁 visible
○ green_light <i>BEElement</i>	9531392		👁 visible

> Step 2 : Modify the element

- ▶ I need a widget to modify the selected element
- ▶ I need to know wich element are selected
- ▶ I need a system that will use the DSL to change the element



A Traffic Light example (2/3)

Plugin = a tool added to the editor

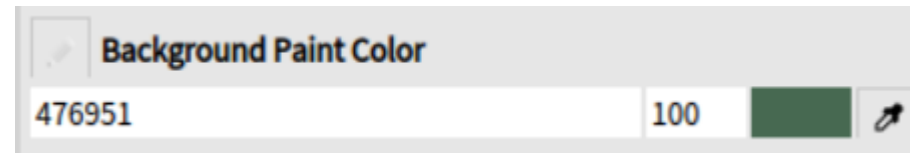
> Tree plugin

- ▶ To select one element and display the selection

Elements	Hash	z	Visibility
▼ light_frame <i>BElement</i>	314609408		visible
○ red_light <i>BElement</i>	100126694		visible
○ yellow_light <i>BElement</i>	425302528		visible
○ green_light <i>BElement</i>	9531392		visible

> Background plugin

- ▶ To Display and change the color of the selected elements



A Traffic Light example (3/3)

The screenshot shows a CAD software interface with a traffic light model. The main window displays a traffic light with three circular lights (red, yellow, green) arranged vertically. The green light is highlighted with a dashed yellow border, and its dimensions are shown as 50.0 in both width and height. The interface includes a top toolbar with icons for file operations, a left sidebar with a 'Tree' view, and a right sidebar with 'Visuals' and 'Layout' tabs. The 'Tree' view shows a hierarchy of elements: 'light_frame BElement', 'red_light BElement', 'yellow_light BElement', and 'green_light BElement'. The 'green_light BElement' is selected. The 'Visuals' tab in the right sidebar shows the 'Background Paint Color' property set to a dark green color with a value of 476951 and 100. Other properties like 'Element Outskirts', 'Border Opacity', 'Border Width', and 'Border Dash Array' are also visible.

Elements	Hash	z	Visibility
light_frame BElement	314609408		visible
red_light BElement	100126694		visible
yellow_light BElement	425302528		visible
green_light BElement	9531392		visible

A plugin based architecture

> System to add widgets to the view

> System to select the element

> System to modify the element

> Plugins can

‣ Add new **widgets** to the view of the editor

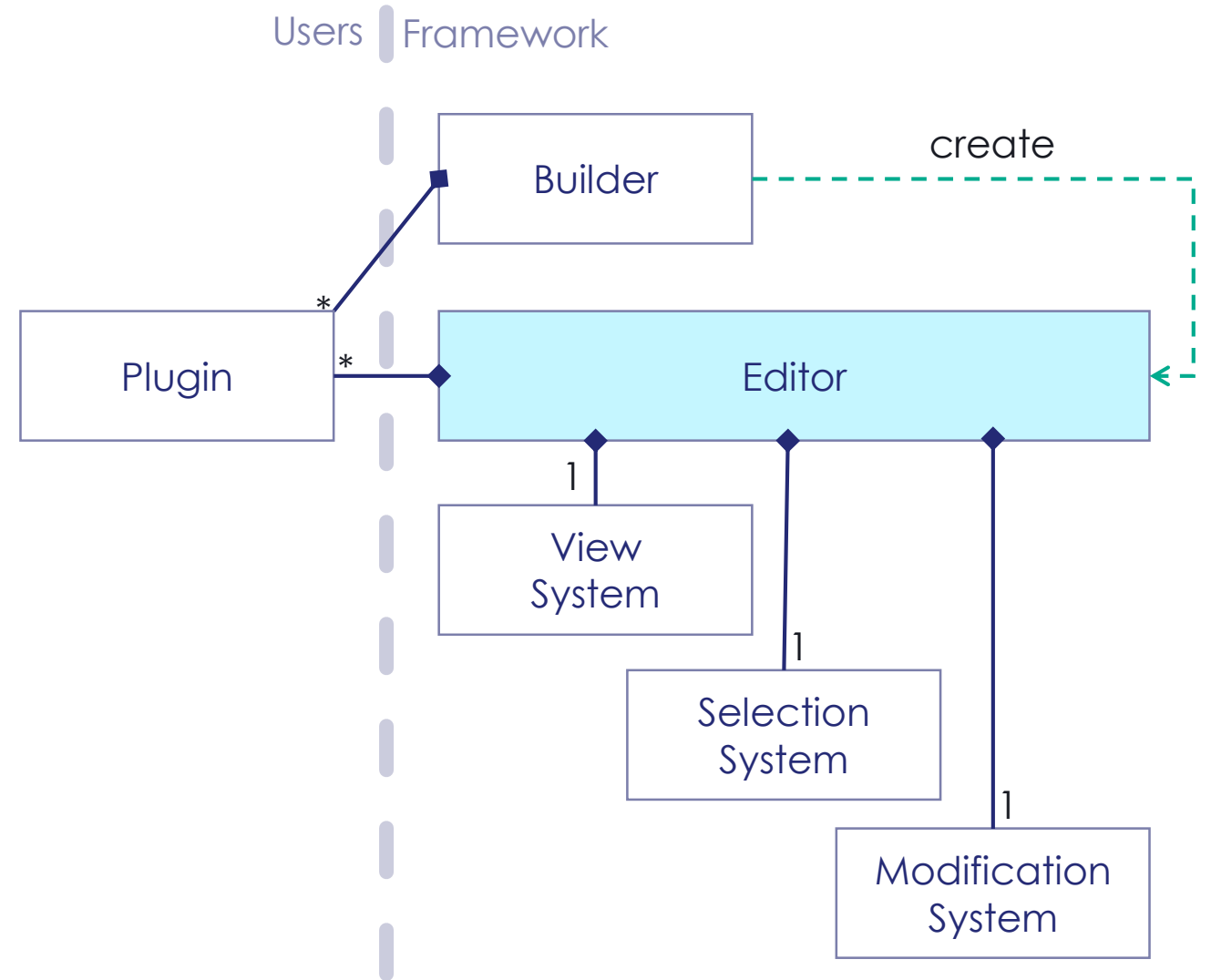
‣ Modify the current selection of the editor

‣ Add new **properties** to the modification system

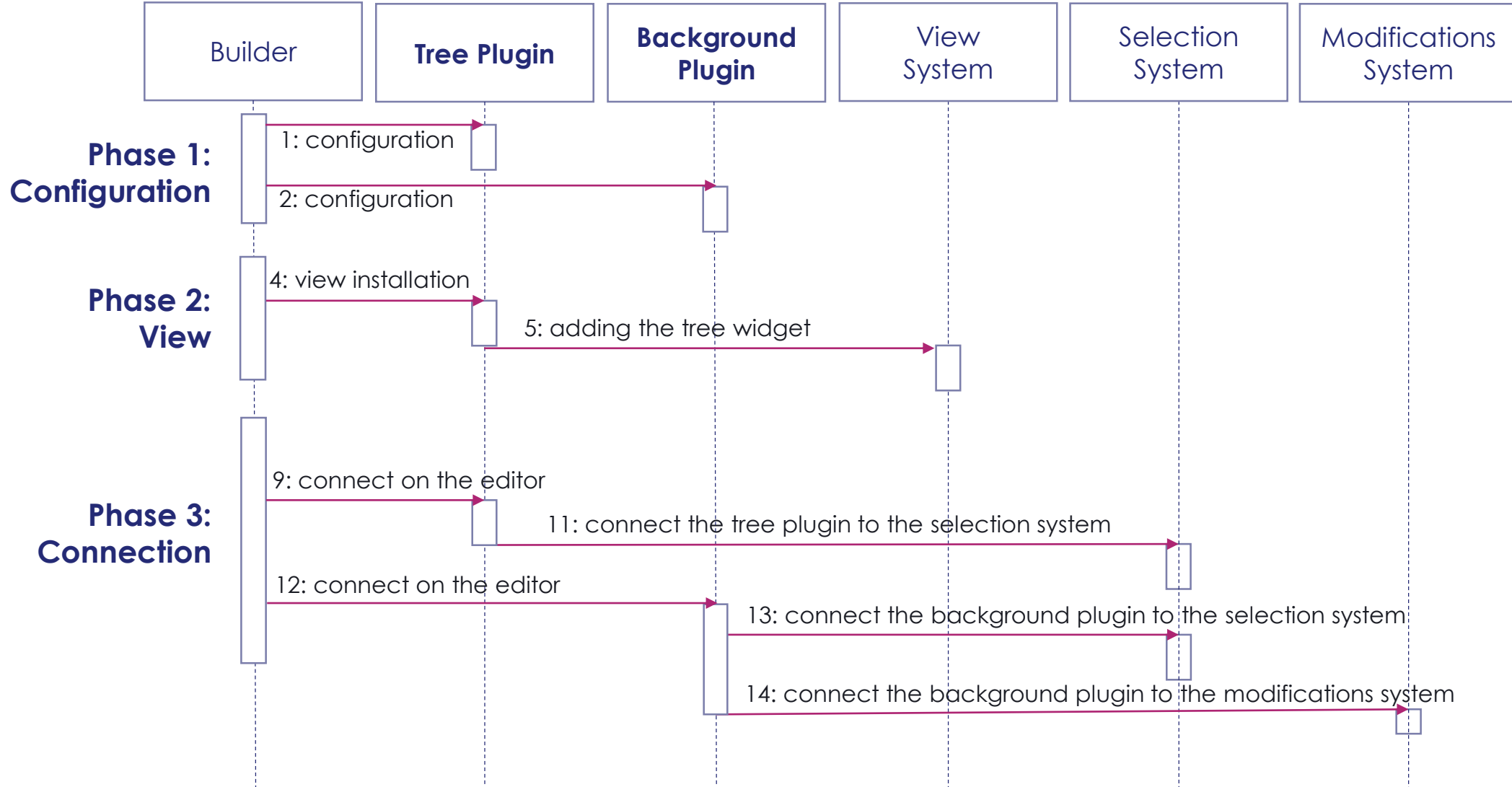
‣ Interract with other plugins

> **Builder to create the Editor**

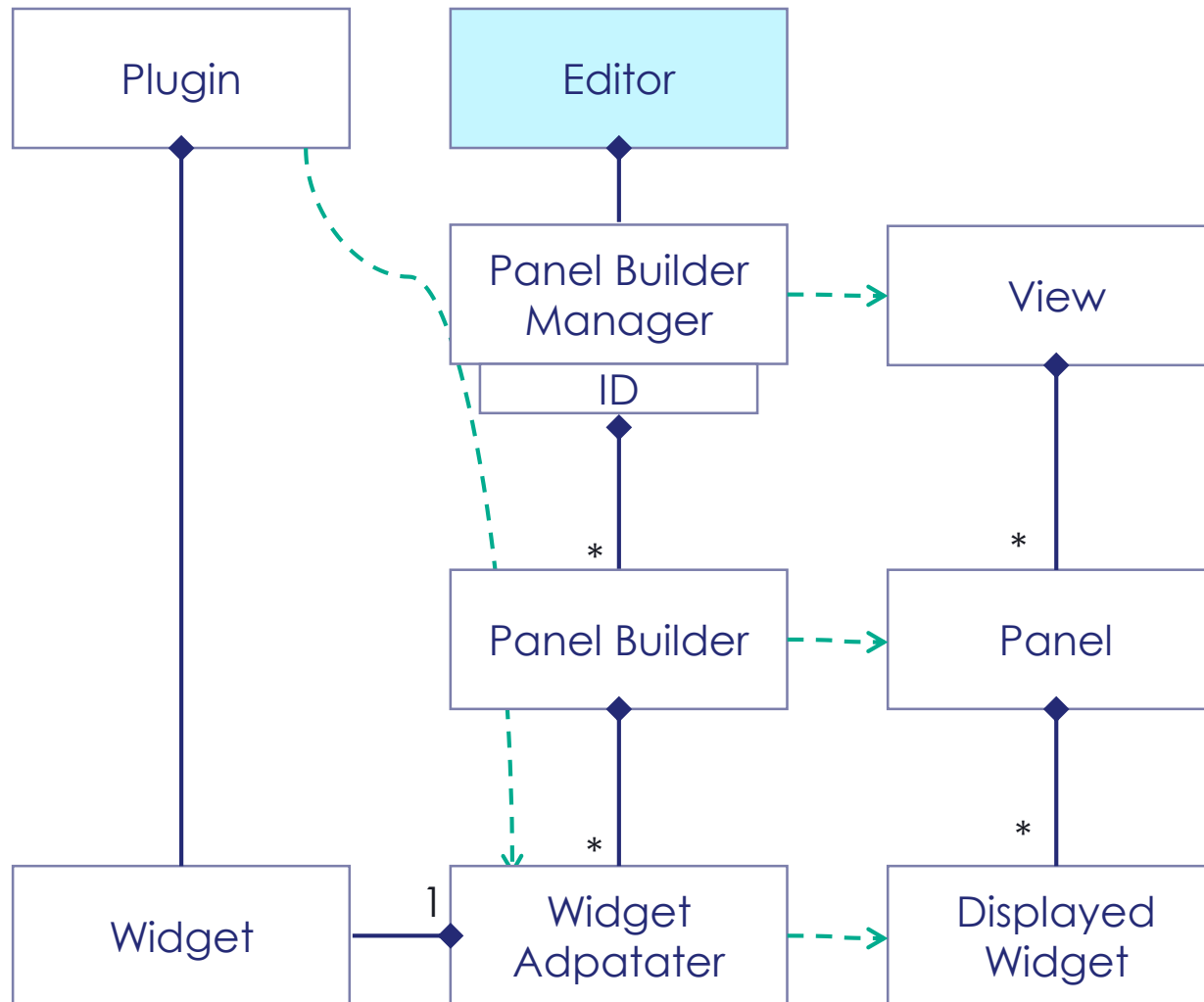
‣ It uses a list of plugins and **install** them on the editor



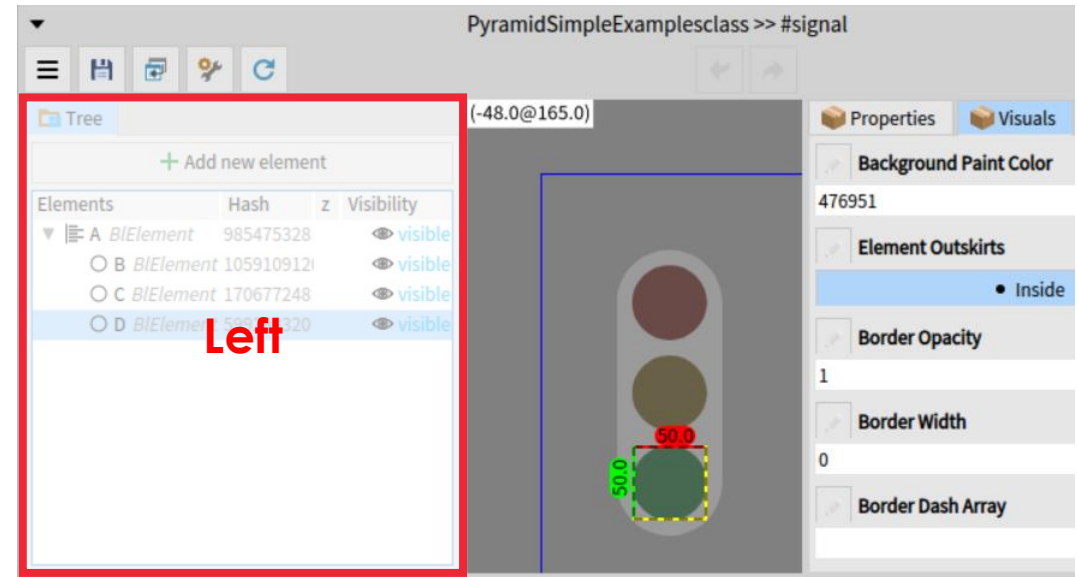
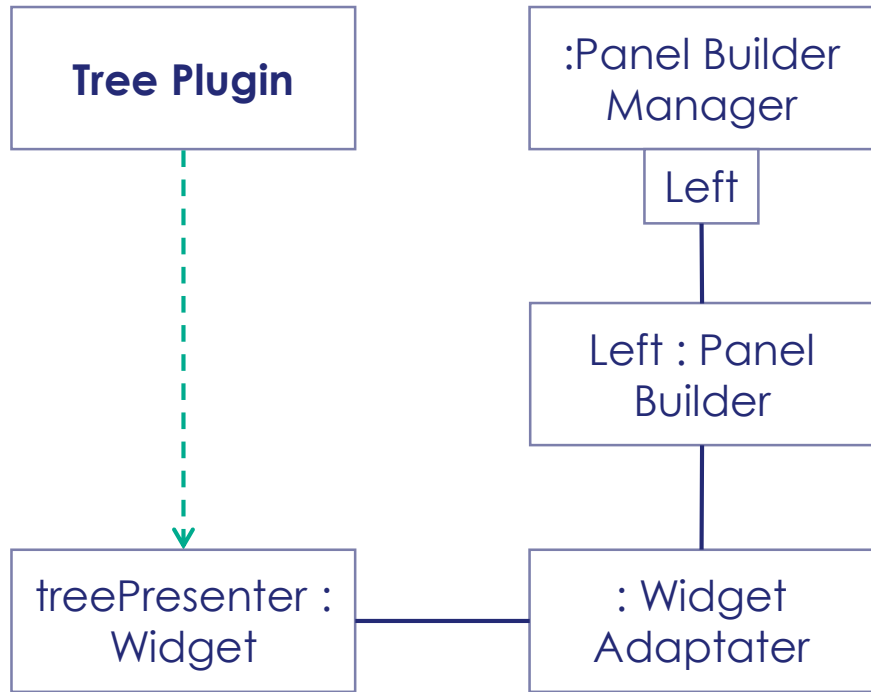
The installation of the plugins



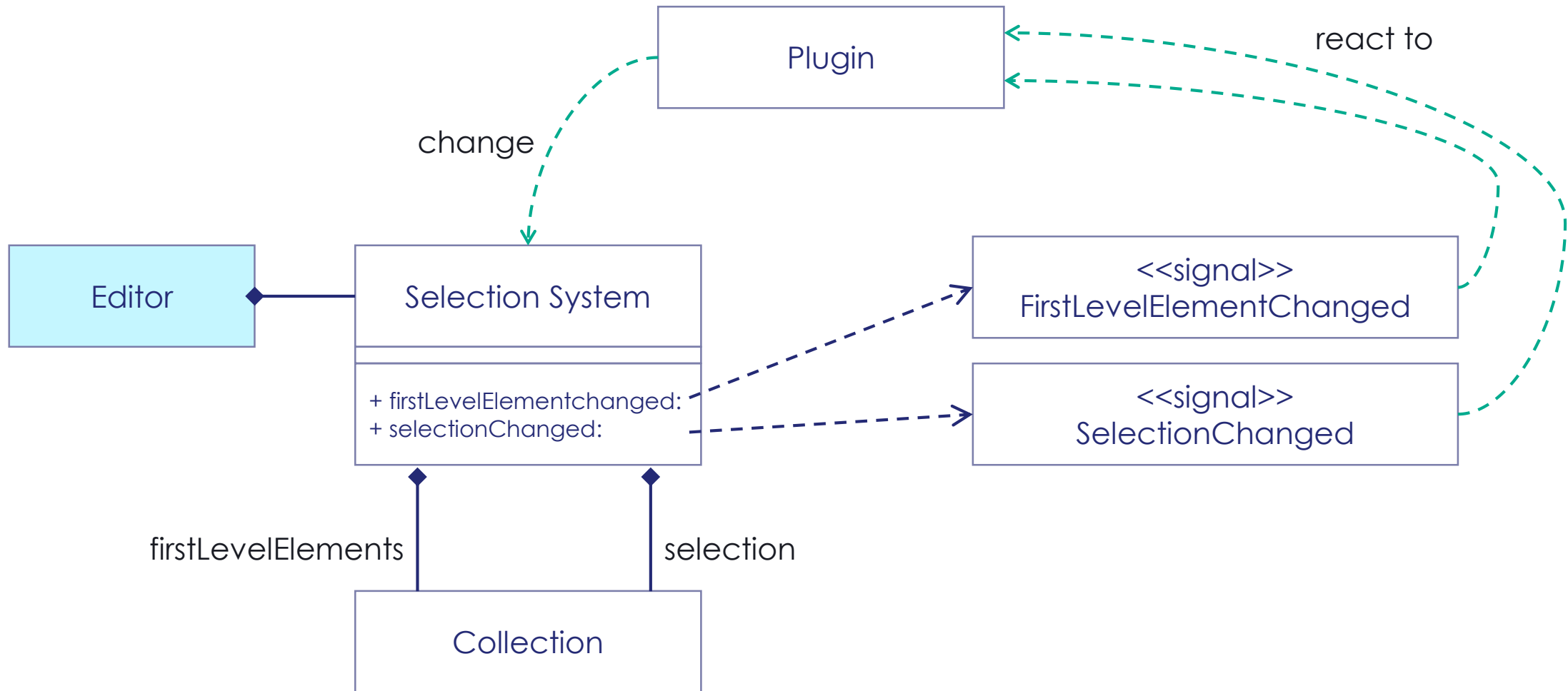
View System (1/2)



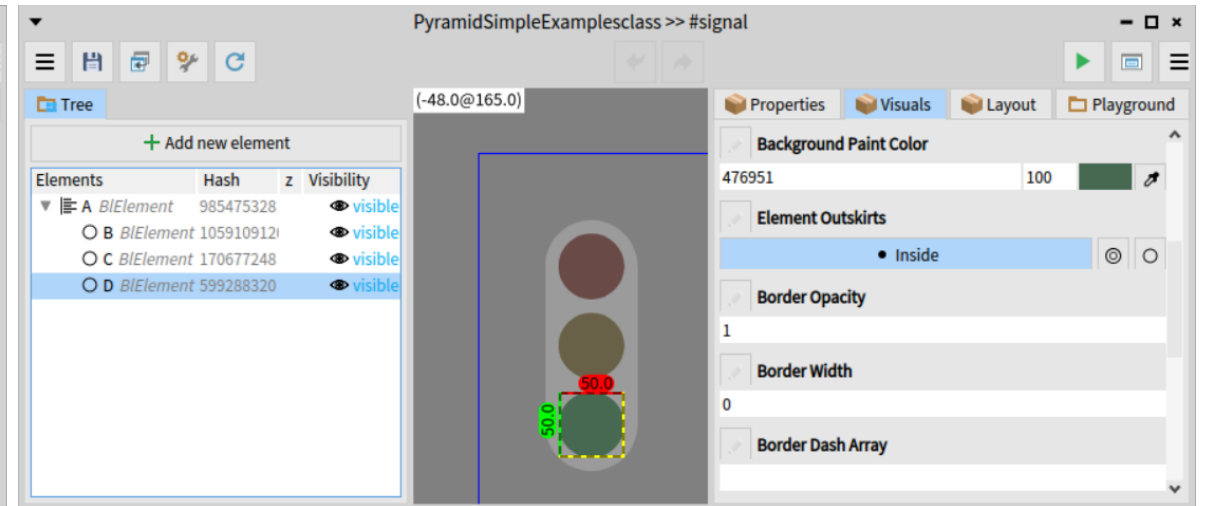
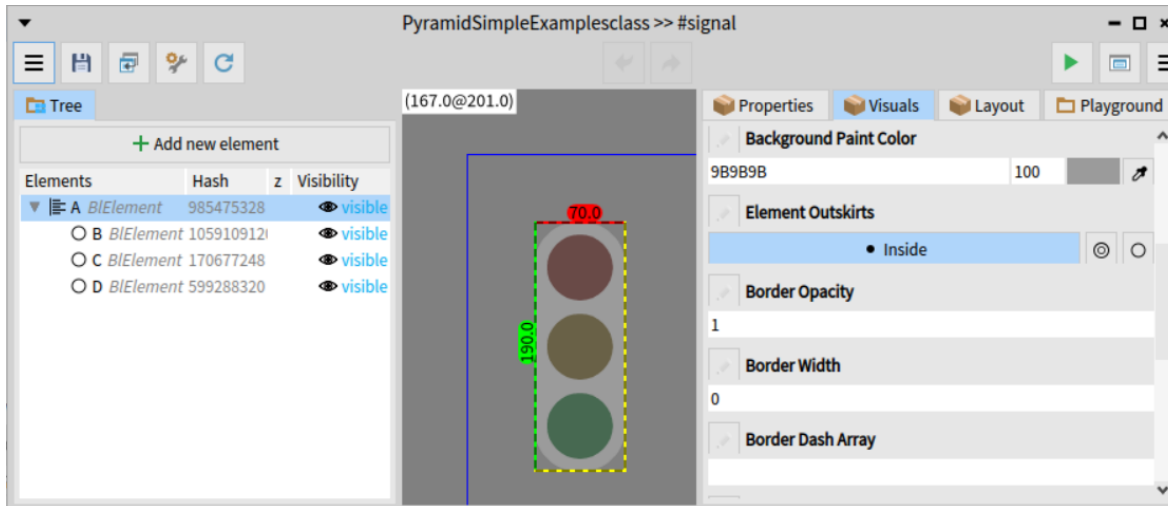
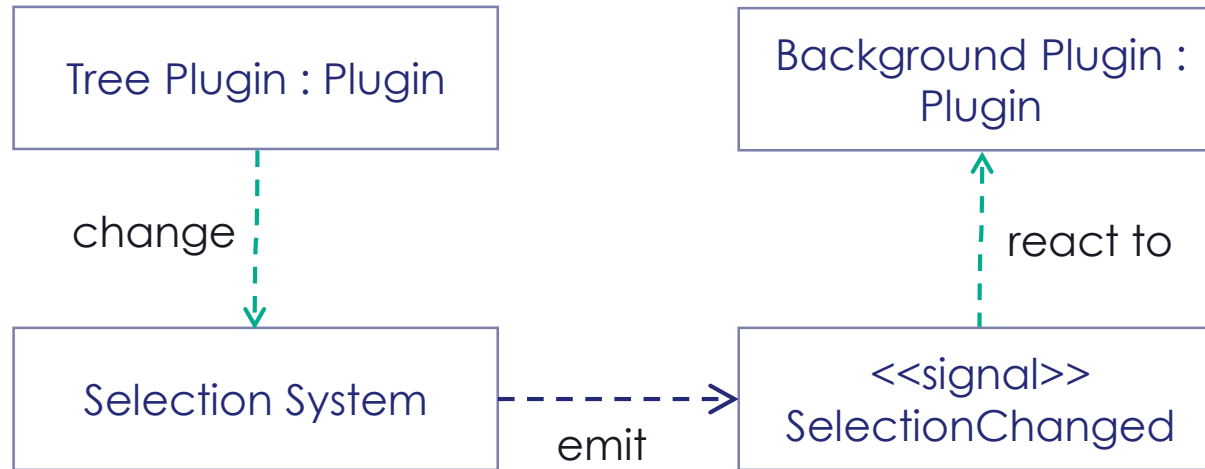
View System (2/2)



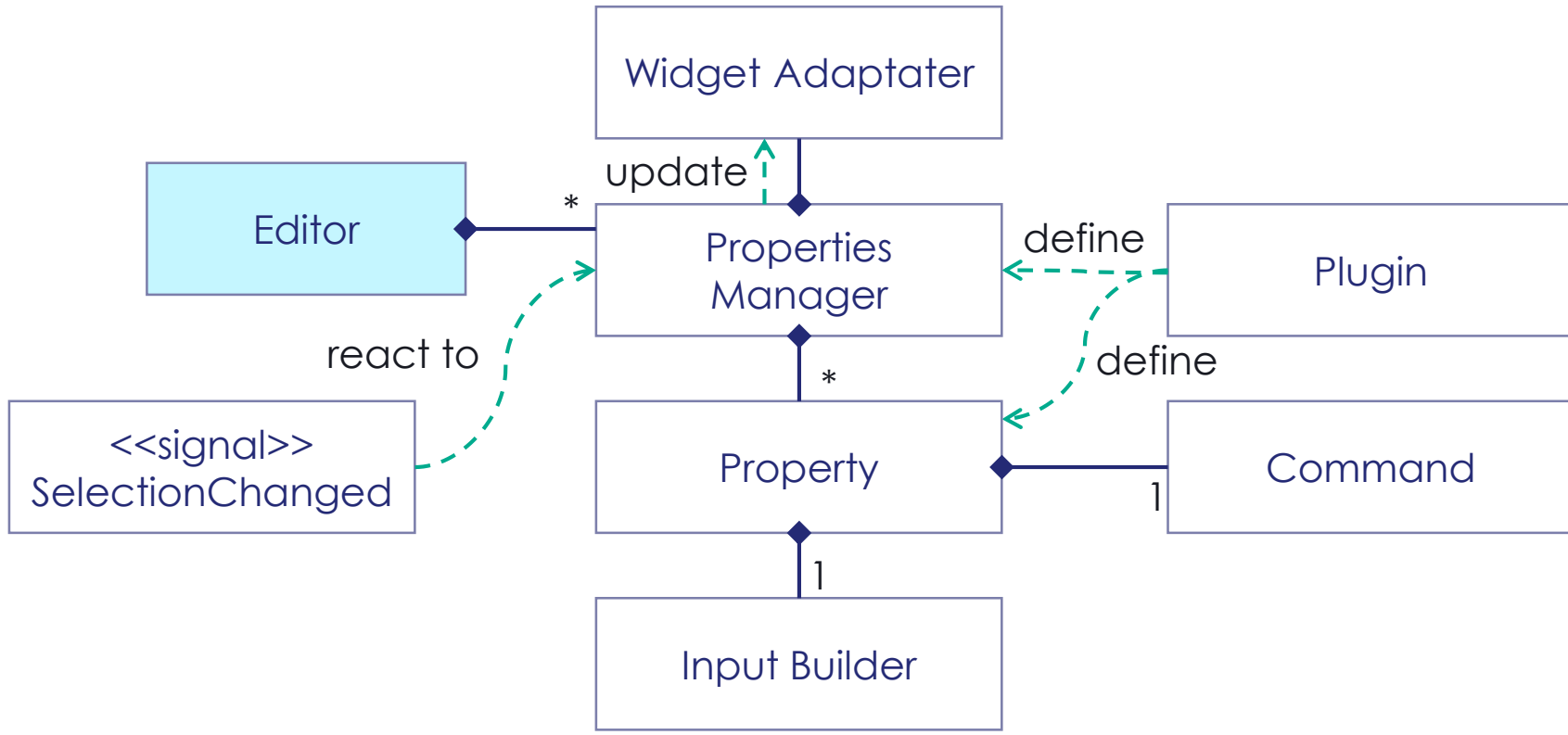
Selection System (1/2)



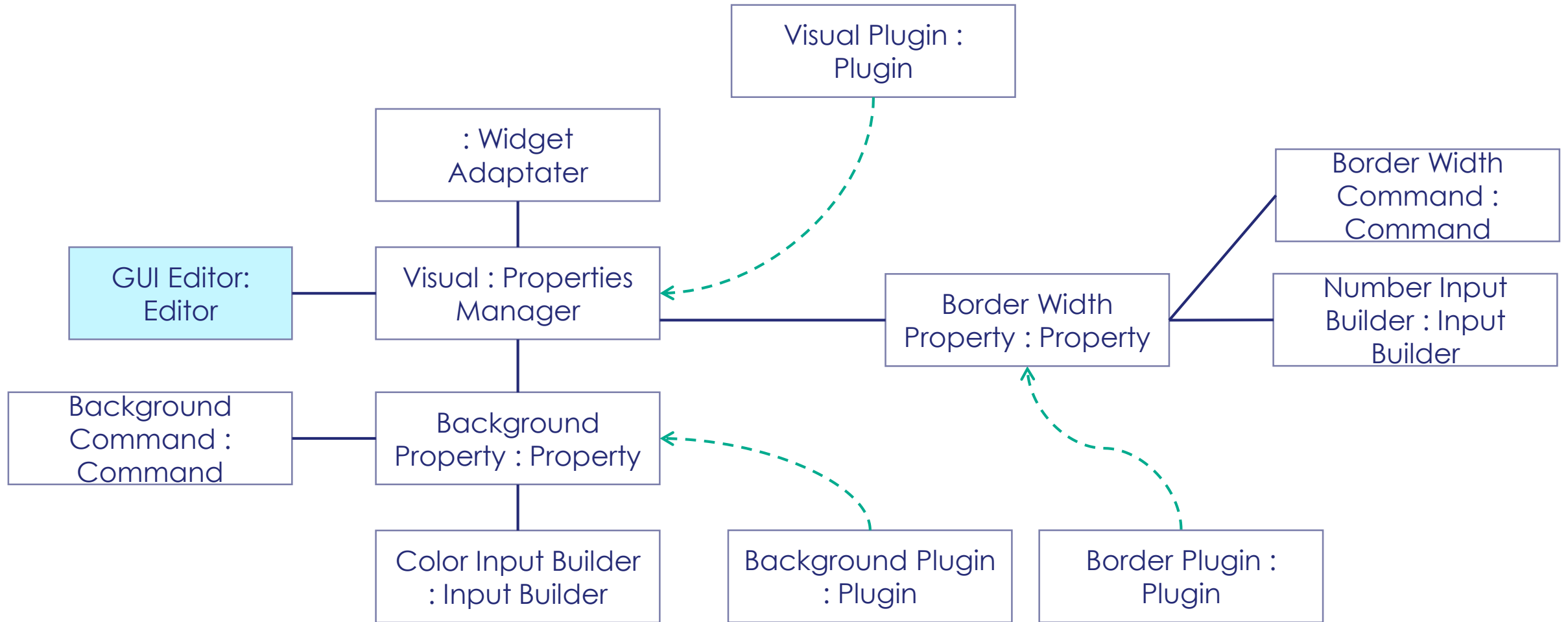
Selection System (2/2)



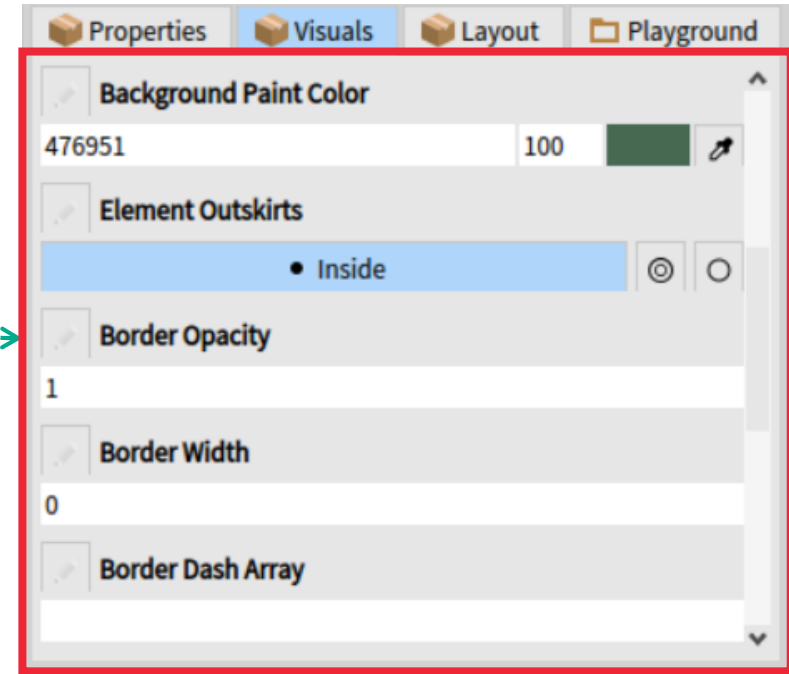
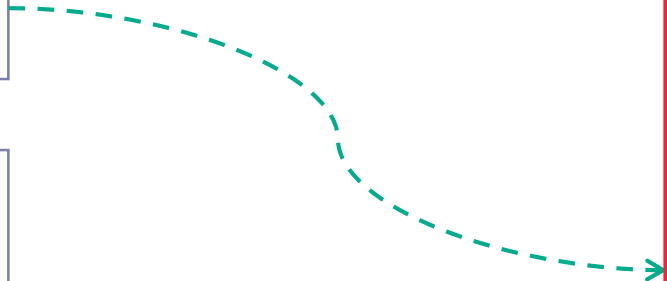
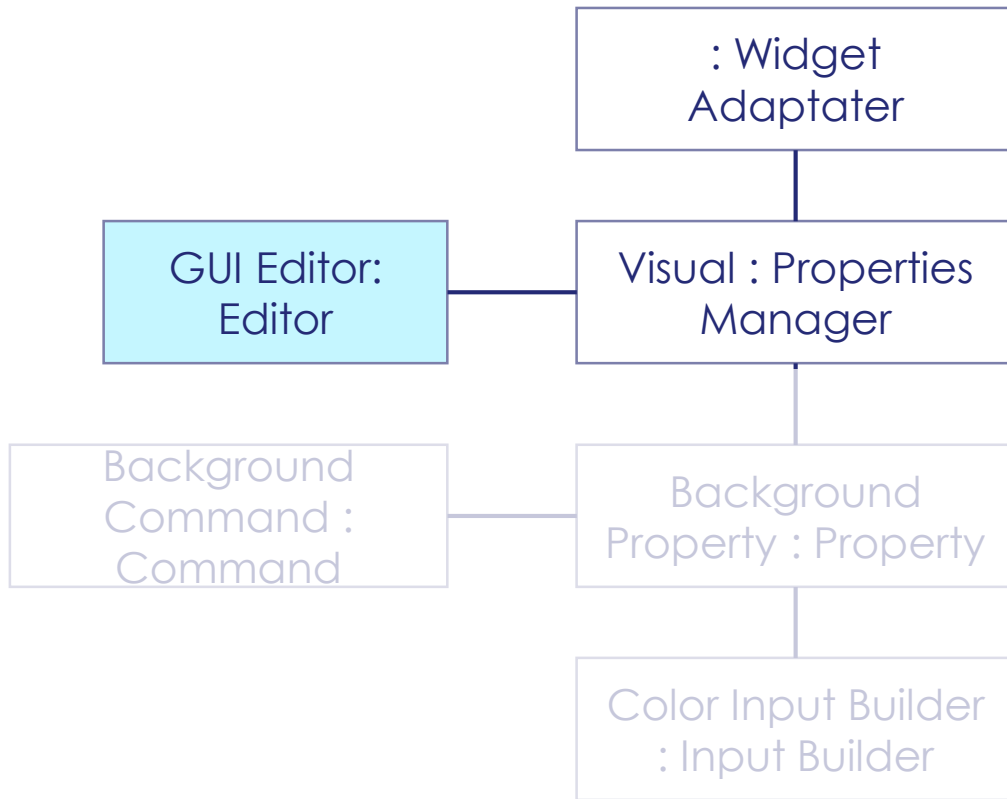
Modifications System (1/4)



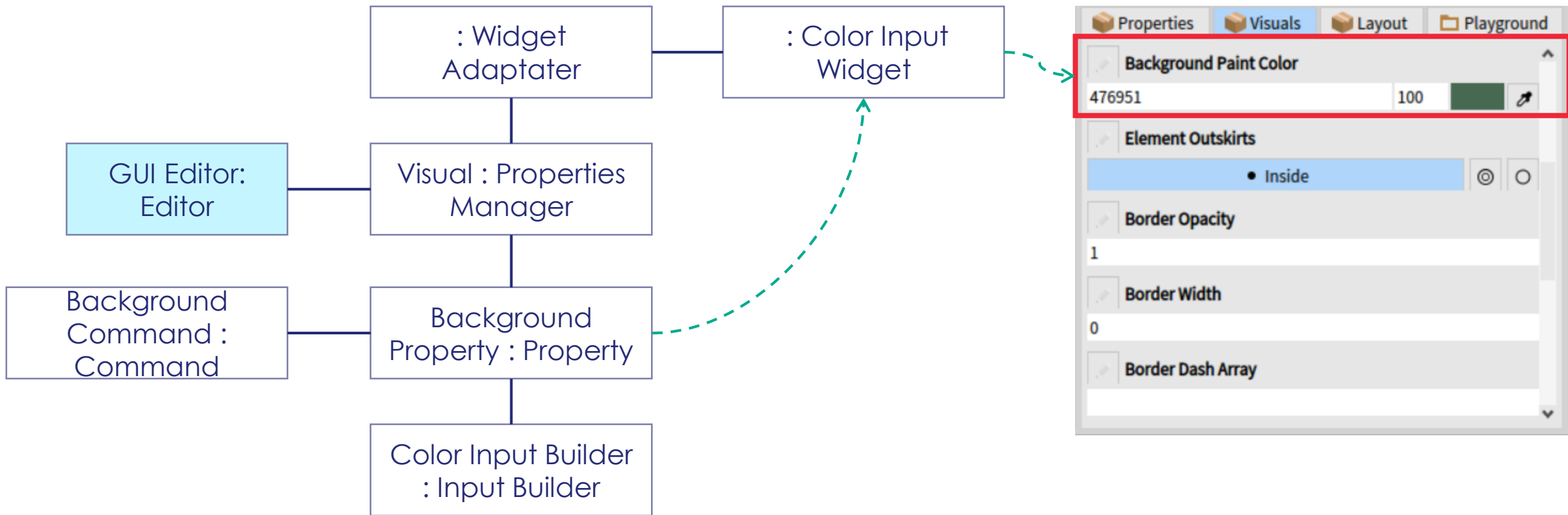
Modifications System (2/4)



Modifications System (3/4)



Modifications System (4/4)



Multi Selection

Single Selection

Elements	Hash	z	Visibility
<input checked="" type="radio"/> A <i>BlElement</i> 378542592			visible
<input type="radio"/> B <i>BlElement</i> 157239296			visible
<input type="radio"/> C <i>BlElement</i> 535808512			visible
<input type="radio"/> D <i>BlElement</i> 292089856			visible
<input type="radio"/> E <i>BlElement</i> 899955200			visible



Background Paint Color		
D13983	100	

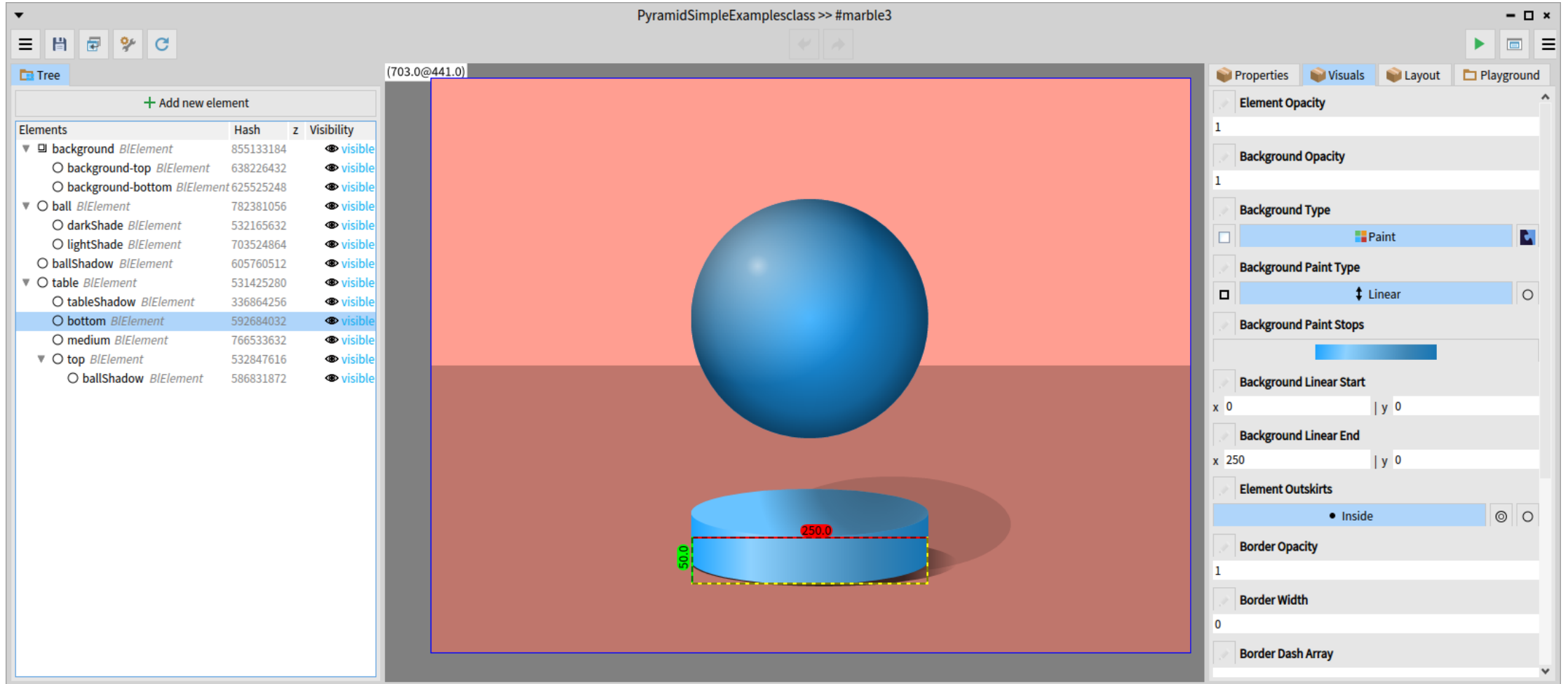
Multi Selection

Elements	Hash	z	Visibility
<input type="radio"/> A <i>BlElement</i> 378542592			visible
<input type="radio"/> B <i>BlElement</i> 157239296			visible
<input type="radio"/> C <i>BlElement</i> 535808512			visible
<input type="radio"/> D <i>BlElement</i> 292089856			visible
<input checked="" type="radio"/> E <i>BlElement</i> 899955200			visible



Background Paint Color		<input type="radio"/> Group	<input checked="" type="radio"/> Individual
899955200	6C4E9B	100	
292089856	6C4E9B	100	
378542592	D13983	100	
157239296	D13983	100	
535808512	D13983	100	

Pyramid: a Bloc editor created with Pyramidion at Thales



THALES

Building a future we can all trust

UBO

Université de Bretagne Occidentale

Lab-STICC

UMR 6285

Merci

www.thalesgroup.com

www.univ-brest.fr

www.labsticc.fr