

SwapView

INHERITS FROM	View:Responder : Object
DECLARED IN	"SwapView.h"

CLASS DESCRIPTION

SwapView is a class that provides all the necessary functionality to swap out several views (ie. inspectors). This is a simple class that is intended to allow the common programmer to create inspectors quickly without a huge expense. This view will swap any panel's content view into itself and then put it back. It stores none of the inspector panels as some other views of this nature do. To do that in my opinion is to limit the functionality. Instead this is a class that will cover all the areas where one area needs to display more than one view. Also other implementations limit the views to only those registered to them. This allows 'dynamic' (meaning non-static) swapping of views (ie. you could dynamically load a class with another panel and have no problem swapping to it).

When you want to swap to a view, simply send a - swapIt call to the SwapView. It in turn will ask its delegate - whatPanel. The delegate will return the panel id and that panel's content view will be swapped in. If the delegate responds with NULL to the -whatPanel call then the SwapView will 'erase' itself by filling its bounds with NX_LTGRAY.

INSTANCE VARIABLES

<i>Inherited from Object</i>	Class	isa;
<i>Inherited from Responder</i>	id	nextResponder;
<i>Inherited from View</i>	NXRect	frame;
	NXRect	bounds;
	id superview;	
	id subviews;	
	id window;	
	struct __vFlags {	
	unsigned int	noClip:1;
	unsigned int	translatedDraw:1;
	unsigned int	drawInSuperview:1;
	unsigned int	alreadyFlipped:1;
	unsigned int	needsFlipped:1;
	unsigned int	rotatedFromBase:1;
	unsigned int	rotatedOrScaledFromBase:1;
	unsigned int	opaque:1;
	unsigned int	disableAutodisplay:1;

```

        unsigned int    needsDisplay:1;
        unsigned int    validGState:1;
        unsigned int    newGState:1;
    }    vFlags;

```

Declared in SwapView

```

id    lastInspector;
id    currentInspector;
id    delegate;
float    backgroundGray;

```

lastInspector

This is the id of the last panel swapped in. SwapView will make sure that you are not swapping the same panel in when its content view is already the swap view.

currentInspector

This is the id of the current panel. That panel's contentView is the current SwapView contentView.

delegate

This is the id of the delegate who is expected to answer the call - whatPanel.

METHOD TYPES

Initializing and freeing View objects

- initWithFrame:
- init

Objects and Delegate Methods	<ul style="list-style-type: none"> - free - delegate - setDelegate: - currentInspector - lastInspector
Background Gray Methods	<ul style="list-style-type: none"> - backgroundGray - setBackgroundGray:
Swapping out views	<ul style="list-style-type: none"> - swapIt: - swapIt

INSTANCE METHODS

currentInspector

- **currentInspector**

Returns the id of the panel. It is that panel's content view which is currently the visible view in the SwapView. If this returns NULL, there is no current panel, and the view should be NX_LTGRAY.

See also: - **lastInspector**

backgroundGray

- **(float)backgroundGray**

Returns a float value which will be used as the gray to erase the view when the delegate returns a NULL id to the call -whatPanel. The default is NX_LTGRAY.

See also: - **setBackgroundGray:**

delegate

- **delegate**

Returns the id of the current delegate. The delegate should respond to - whatPanel.

See also: - **setDelegate:**

init

- **init**

Returns [self initWithFrame:defaultFrame]. The default frame is {{0.0, 0.0}, {0.0, 0.0}}.

See also: - **initWithFrame:**

initWithFrame

- **initWithFrame:(NXRect *)frame**

Sets all instance vars to NULL and calls super - initWithFrame:frame.

See also: - **init**

lastInspector

- **lastInspector**

Returns the id of the panel where the current contentView of SwapView came from.

See also: - **currentInspector**

setBackgroundGray

- **setBackgroundGray:(float)aGray**

Sets the float value which will be used as the gray to erase the view when the delegate returns a NULL id to the call -whatPanel. The default is NX_LTGRAY.

See also: - **backgroundGray**

setDelegate

- **setDelegate:anObject**

Sets the delegate to anObject. The delegate must respond to - whatPanel for the -swapIt call to swap out to another view.

See also: - **delegate**

swapIt

- **swapIt**

This is the heart of this class. This method calls the delegate with - whatPanel. It then removes the current subview of SwapView, if there is one (ie lastInspector). Then it installs the offscreen panel's contentView as the currentInspector, and sends it - display. If the delegate responds with NULL or there is no delegate, then the SwapView is NX_LTGRAY. Returns self.

See also: - **swapIt:**

swapIt:

- **swapIt:sender**

Sender is not used for anything. This simply calls - swapIt.

See also: - **swapIt**

METHODS IMPLEMENTED BY DELEGATE

whatPanel

- **whatPanel**

This should return the id of the panel who's contentView should be swapped into the SwapView on screen. Most of the time this panel should be offscreen. It is important to set up the panel so that it is not deferred. If it is deferred, - swapIt will not be able to allocate a gstate for the contentView and therefore will not be able to swap it onto the screen.

See also: - **delegate**, - **setDelegate**