

HotShape

INHERITS FROM	Control : View : Responder : Object
DECLARED IN	HotShape.h
ADDITIONAL FILES REQUIRED	HotShape.psw, SmallCell.h
WRITTEN BY	Charles G. Fleming

CLASS DESCRIPTION

A HotShape is a Control subclass that intercepts mouse-down events and sends an action message to a target object whenever the mouse is clicked in the hot region. The hot region is determined by a path in the associated file HotPath.psw. The hot path can be visible or invisible.

The standard HotPath.psw file contains, in the function setHotPaths(), the postscript for elliptical, rectangular, diamond, star, triangular and octagonal paths. If additional paths are desired for your HotShape control, simply add the postscript to the setHotPaths() function in HotPath.psw. If you are using the HotShape palette (see below), you will need to recompile the palette.

For ease of use, the HotShape has been palettized and has an inspector which allows you to select the hot path for the HotShape, whether the hot shape is visible or invisible, and to set the tag. Any project built with the HotShape palette will need to add HotShape.(hm), SmallCell.(hm) and HotShape.psw to the project being built.

INSTANCE VARIABLES

<i>Inherited from Object</i>	Class	isa;
<i>Inherited from Responder</i>	id	nextResponder;
<i>Inherited from View</i>	NXRect NXRect id superview; id subviews; id window; struct __vFlags	frame; bounds; vFlags;
<i>Inherited from Control</i>	int id cell; struct _conFlags	tag; conFlags;
<i>Declared in HotShape</i>	char * BOOL	hotPathName; visible;

hotPathName

The name of the current hot path.

visible

Whether or not the current hot path is visible.

METHOD TYPES

Initializing a HotShape ± initWithFrame:

Aiding event handling

- acceptsFirstMouse
- mouseDown:

Modifying graphic attributes

- visible
- isVisible

Setting the hot path

- setHotPathName:
- hotPathName

Modifying the frame rectangle

- sizeTo::

Displaying

- drawSelf::

Working with Interface Builder - inspectorName

Archiving - awake
- read:
- write:

INSTANCE METHODS

acceptsFirstMouse

- (BOOL)**acceptsFirstMouse**

Returns YES. HotShapes always accept the mouse-down (in its hot region) that activates a Window.

awake

- **awake**

Calls the function *setHotPaths()* (defined in HotPath.psw) to define the names of the paths that can be used by the HotShape. Returns **self**.

drawSelf::

- **drawSelf:**(const NXRect *)*rects* :(int)*rectCount*

Draws the hot path in the HotShape if *visible* is YES. The hot path used is determined by the instance variable *hotPathName*. Returns **self**.

hotPathName

- (char *)**hotPathName**

Returns the name of the hot path used by the HotShape.

initWithFrame:

- **initWithFrame:**(const NXRect *)*frameRect*

Initializes and returns the receiver, a new HotShape instance. The HotShape is scaled so that it has unit length and height. The *visible* instance variable is set to YES. A SmallCell is installed as the HotShape's cell. This allows the HotShape to be resized as small as desired. The default HotPath is a diamond shape.

inspectorName

- (const char *)**inspectorName**

Returns "HotShapePaletteInspector".

mouseDown:

- **mouseDown:**(NXEvent *)*theEvent*

If the mousedown and subsequent mouseup both occur in the hot path, then the HotShape's action is sent to its target. Returns **self**.

read:

- **read:**(NXTypedStream *)*stream*

Reads the HotShape and its instance variables *hotPathName* and *visible* from the typed stream *stream*. Returns **self**.

setHotPathName:

- **setHotPathName:**(char *)*name*

Sets the name of the hot path to be drawn into the HotShape. Returns **self**.

setVisible:

- **setVisible:**(BOOL)*seeIt*

Sets whether or not the hot path is visible. Returns **self**.

sizeTo::

- **sizeTo:**(NXCoord)*width* :(NXCoord)*height*

Overrides the inherited **sizeTo::** method to preserve unit width and height. Returns **self**.

visible

- (BOOL)**visible**

Returns whether or not the hot path is visible.

write:

- **write:**(NXTypedStream *)*stream*

Writes the receiving HotShape and its instance variables *hotPathName* and *visible* to the typed stream *stream*. Returns **self**.