

MaskGroup

Table of Contents

- [Description](#)
- [Fields](#)
- [Example](#)

Extends: [Group](#)

Description

The **MaskGroup** node class allows an alpha mask bitmap to be applied to the rendering of its children. This allows effects such as having a list fade out at the bottom to be easily created.

MaskGroup nodes do not work on all Roku players (specifically, **MaskGroup** nodes only work on players that support OpenGL). On players whose graphics do not support OpenGL, a **MaskGroup** node just renders its children without applying the extra alpha mask. For this reason, you should avoid using the **MaskGroup** node for anything other than decorative purposes, such as creating a fading effect at the bottom of the screen, where the user interface information value of the screen element is not affected by the lack of the **MaskGroup** node rendering.

Unlike other bitmaps, the **MaskGroup** node class does not work with 9-patch images. If the bitmap size does not match the group bounding rectangle, the edge rows of the mask are repeated as needed.

For example, suppose the **MaskGroup** node bounding rectangle has an origin at (0, 0), a width of 150 and a height of 80. If the mask bitmap were also 150 by 80, and the `maskOffset` and `maskSize` field values were left at their default values of [0, 0], then each pixel of the mask bitmap would be applied to the corresponding pixel of the **MaskGroup** node bounding rectangle. The figure below illustrates that case. On the right is a **MaskGroup** node that contains a **Poster** node. In the center is an alpha bitmap with the same size as the **Poster** node. On the right is the result of rendering the **MaskGroup** node with the image in the center used as a mask.

The `maskSize` and `maskOffset` field values can be used to scale and offset the mask bitmap relative to the group coordinate system, as shown in the figure below. The mask is the same bitmap as in the figure above. In the below figure, the `maskOffset` field value has been set to (50, 30), offsetting the origin of the mask bitmap coordinate system relative to the origin of the **MaskGroup** node coordinate system 50 pixels to the left and 30 pixels down. The `maskSize` field value has been set to (100, 50). This causes the width of the mask bitmap to be scaled from 150 to 100, and the height of the mask bitmap to be scaled from 80 to 50, so that the mask right edge aligns with the right edge of the **MaskGroup** node bounding rectangle, and the mask bottom edge aligns with the bottom edge of the **MaskGroup** node bounding rectangle. The center of the figure shows the mask with the `maskOffset` and `maskSize` field values applied overlaid on top of the **MaskGroup** node. The right side of the figure shows the result of applying the mask to the **MaskGroup** node. Notice that the left column of pixels in the mask is used as the mask for all columns of the **MaskGroup** node that lie to the left of the mask, causing those columns to be rendered with full opacity. Similarly, and the top row of the mask is used as the mask for all rows of the **MaskGroup** node that lie above the mask, causing each of the rows to fade out on the right.

As shown in above, the transformed mask may not overlay all the rendered pixels of the **MaskGroup** node. In that case, the leftmost (rightmost) columns of the mask bitmap are used as the mask for any **MaskGroup** node pixels that are to the left (right) of the transformed mask. The top (bottom) row of the mask bitmap is used as the mask for **MaskGroup** node pixels that are above (below) the transformed mask.

Fields

Field	Type	Default	Use
-------	------	---------	-----

maskUri	URI string	""	Specifies the bitmap to use for the group alpha mask. The alpha value of each pixel rendered in the group is multiplied by the value of a pixel in the alpha mask. The mapping from pixel coordinates in the group to mask pixel coordinates is controlled by the <code>maskSize</code> and <code>maskOffset</code> field values. Those fields define a coordinate system for the mask bitmap relative to the group coordinate system. This mask coordinate system is used to map mask pixels to their corresponding pixels in the MaskGroup node.
maskSize	vector2d	[0, 0]	Specifies a scaling factor of the alpha mask coordinate system relative to the group coordinate system. If either element of the <code>maskSize</code> field value does not match the corresponding size of the mask bitmap, the mask bitmap is scaled to the specified size in the dimension. Setting either element of the <code>maskSize</code> field vector to 0 causes the mask bitmap to be used at its actual size in that dimension.
maskOffset	vector2d	[0, 0]	Specifies an offset of the mask coordinate system relative to the group coordinate system. For example, if the <code>maskOffset</code> field value is set to [100, 0], then mask pixel [0, 0] is used as the alpha value for pixels rendered for the MaskGroup node coordinate [100, 0].
maskBitmapWidth	float	0	Read-Only Contains the actual width of the mask bitmap. The mask bitmap can be stretched horizontally by setting the x-dimension of the <code>maskSize</code> field value to a different value than the <code>maskBitmapWidth</code> field value.
maskBitmapHeight	float	0	Read-Only Contains the actual height of the mask bitmap. The mask bitmap can be stretched vertically by setting the y-dimension of the <code>maskSize</code> field value to a different value than the <code>maskBitmapHeight</code> field value.

Example

[MaskGroup.zip](#) shows using a **MaskGroup** node to conceal and reveal a poster.