IWEBGLTEXT INTERFACE

View online: https://www.construct.net/en/make-games/manuals/addon-sdk/reference/graphics-interfaces/iwebgltext

The `IWebGLText` interface manages text wrapping, drawing text to a canvas, and then uploading the result to a WebGL texture. This makes it easy to display text in a WebGL renderer. It is created via the IWebGLRenderer `CreateWebGLText()` method.

Methods

Release()

Destroy the object and its resources. `IWebGLText` must be released when it is no longer needed; do not simply drop references, otherwise not all of its resources will be collected. If your plugin creates an IWebGLText, it should release any it still uses in its own `Release()` method.

SetFontName(name)

Set the name of the font face used for drawing text.

SetFontSize(ptSize)

Set the size of the font, in points, used for drawing text.

SetLineHeight(px)

Set the extra line height spacing, in pixels, used for drawing text. Note 0 is the default, indicating no offset to the default line height.

SetBold(b)

Set the bold flag used for drawing text.

SetItalic(i)

Set the italic flag used for drawing text.

SetColor(color)

Set the color of the text using a SDK.Color or a string. If a string is passed, it is passed directly to a 2D canvas `fillStyle` property, so can be anything that property accepts, e.g. "red", "#00ffee", "rgb(0, 128, 192)" etc.

SetColorRgb(r, g, b)

Set the color of the text using separate RGB components.

SetHorizontalAlignment(h)

Set the horizontal alignment of the text within its bounding box. This can be one of `"left"`, `"center"` or `"right"`.

SetVerticalAlignment(v)

Set the vertical alignment of the text within its bounding box. This can be one of `"top"`, `"center"` or `"bottom"`.

SetWordWrapMode(m)

Set the word wrapping mode. This can be one of `"word"` (for space-delimited word wrapping) or `"character"` (for wrapping on any character).

SetText(text)

Set the text string to be drawn.

SetSize(width, height, zoomScale)

Set the size of the area that text can be drawn in. The size is specified in CSS pixels. The `zoomScale` can be increased to render the text at a higher resolution, which is useful when zooming in the Layout View.

GetTexture()

Get an IWebGLTexture interface representing the texture with the requested text rendered on to it. **Note:** the texture is generated asynchronously, so can return `null` when first requested. Use `SetTextureUpdateCallback()` to get a callback when the texture has updated, where the relevant Layout View can be redrawn to render with the updated texture.

GetTexRect()

Return a SDK.Rect representing the content area of the text on the WebGL texture. This is the subset of the texture that ought to be rendered. Note: this is only valid when `GetTexture()` returns a non-null result.

SetTextureUpdateCallback(callback)

Set a function to call when the texture containing the rendered text is updated. Since the texture is generated asynchronously, this is necessary to know when to redraw any views that may be displaying the text, so they can redraw with the updated texture.

ReleaseTexture()

Release the underlying WebGL texture. This can be used to save memory. However the texture will be re-created the next time `GetTexture()` is called.

GetTextWidth()

GetTextHeight()

Return the size of the text bounding box after processing word wrap. This allows determining the size of the actual visible text, rather than the box used for word wrap bounds.