TextEngine

Glyph Spacing

Glyph Spacing is available on TextBrush, and LayoutingOptions. This allows a user to correct the spacing between glyphs, by adding a specified number of pixels to the spaces between glyphs.

BitmapFont Engine

BitmapFont Engine is available for usage inside Candera-based applications. A bitmap font is essentially an image file which consists of several characters images and a header that depicts the size and location of each character inside the image. Each bitmap font might contain several fonts - mainly one font face with multiple sizes. One advantage of utilizing bitmap fonts is that the rendering to the screen requires very little resources and since each character is represented as a sub-texture, they can be reused without requiring additional memory on the GPU.

In order to use BitmapFonts, the CMake variable **CANDERA_FONTENGINE** should be changed to *BitmapFont*.

> Harbuzz Text Shaper cannot be used with BitmapFont, either ComplexScriptLib or NoShaping should be chosen.

This font engine uses a proprietary format to describe glyph bitmaps and glyph meta info. In order to use a bitmap font, provide an implementation for Candera::TextRendering::FontStore.

Bitmap fonts require memory mapped assets. So, when the faces are loaded in be sure the storageType is set correctly:

```
TextRendering::FontResourceDescriptor
descriptor.storageType = TextRendering::FontResourceDescriptor::MemoryResource;
```

Generating Bitmap Fonts

The following steps need to be considered:

- Install Bitmap Font Generator from http://www.angelcode.com/products/bmfont/

Open the application. From the Options menu:

- Choose Font Settings. Set as needed the settings for "Font", "Add font file", "Size" etc.
- Configure the Export Options. Set "Font descriptor" to "XML" and Textures to "png - Portable Network Graphics"
- "Save bitmap font as ..."

Convert the resulted XML and PNG files to binary format using the internal tool BmFontCreator using the following command:

- BmFontCreator. exe -o MyFont32.fbm *.fnt

> A bitmap font should be used at its native pixel size. However, they are scalable because of the use of bitmaps, but only scaling down is recommended. In case of up scaling, the visual quality will be reduced. Including separate font sizes for bitmap fonts is possible in order to achieve the best looking results.

Glyph Substitution

A substitution character may be set within the style with the interface Style::SetDefaultCodepoint. This character is used by the text engine in any of the following situations:

- when a glyph is missing from the style. i.e. no matching font can provide that glyph. Exception make control character, like newline or LRE.
- when UTF8 buffers contain malformed characters.

Character Glyph Mapping

GlyphBitmap now contains information about the position within the text from where the glyphs originated, and the location of the glyph within the font.

Glyph Order

The order in which glyphs are generated may be controlled by setting a property within the TextRenderContext. See TextRenderContext::GlyphOrder for details. This is most relevant when doing preprocessing. OriginalGlyphOrder is intended for use when intermediate measurements of the text are required, for instance when truncating texts. OrigitnalChunkOrder is more performant

then RenderOrder, and is intende for measurements, where the order of the chunk is not important, for instance instance when computing layout rectangles. RenderOrder shall be used when rendering.

Text Preprocessing

The text engine allows text to be preprocessed by providing utility the class PreprocessingContext. This interface is satisfied by two concrete instances.

- MinimalPreprocessingContext - stores all the information need to be able to render the glyphs.
- MaximalPreprocessingContext - stores all the information available within the text renderer for each glyph. TextBrush allows setting PreprocessedText instead of Text, to speed up text processing within the brush.

Revision #4
Created 2023-03-02 07:41:49 UTC by Kanai Tomoaki
Updated 2024-07-29 06:24:09 UTC by Tsuyoshi.Kato