

hgl868的专栏

目录视图 摘要视图 RSS 订阅

个人资料



hgl868



访问： 955278次
积分： 12367
等级：
排名： 第671名

原创： 303篇
转载： 16篇
译文： 1篇
评论： 130条

文章搜索

文章分类

- c++ (84)
- Android (137)
- java (8)
- linux (5)
- Design Pattern (36)
- other (19)
- OpenGL (28)
- games (1)

文章存档

- 2016年05月 (2)
- 2016年04月 (5)
- 2016年03月 (1)
- 2015年11月 (2)
- 2015年08月 (1)

展开

阅读排行

- GLSL 内建函数 (25386)
- NDK Build 参数 (21848)

FreeType 管理字形

标签： vector string image matrix postscript stream

2012-02-13 13:44 5133人阅读 评论(0) 收藏 举报

分类： c++ (83)

版权声明：本文为博主原创文章，未经博主允许不得转载。

1.字形度量

顾名思义，字形度量是对应每一个字形的特定距离，以此描述如何对文本排版。

通常一个字形有两个度量集：用来排版水平文本排列的字形（拉丁文、西里尔文、阿拉伯文、希伯来文等等）和用来排版垂直文本排列的字形（中文、日文、韩文等等）。

要注意的是只有很少的字体格式提供了垂直度量。你可以使用宏FT_HAS_VERTICAL测试某个给出的face对象是否包含垂直度量，`FT_FACE_FLAG_VERTICAL`表示包含垂直度量。

每个的字形度量都可以先装载字形到face的字形槽，然后通过face->glyph->metrics结构访问，其类型为FT_Glyph_Metrics。我们将在下面详细讨论它，现在，我们只关注该结构包含如下的字段：

- Width**
这是字形图像的边框的宽度。它与排列方向无关。
 - Height**
这是字形图像的边框的高度。它与排列方向无关。千万不要把它和FT_Size_Metrics的height字段混淆。
 - horiBearingX**
用于水平文本排列，这是从当前光标位置到字形图像最左边的边界的水平距离。
 - horiBearingY**
用于水平文本排列，这是从当前光标位置（位于基线）到字形图像最上边的边界的水平距离。
 - horiAdvance**
用于水平文本排列，当字形作为字符串的一部分被绘制时，这用来增加笔位置的水平距离。
 - vertBearingX**
用于垂直文本排列，这是从当前光标位置到字形图像最左边的边框的垂直距离。
 - vertBearingY**
用于垂直文本排列，这是从当前光标位置（位于基线）到字形图像最上边的边框的垂直距离。
 - vertAdvance**
用于垂直文本排列，当字形作为字符串的一部分被绘制时，这用来增加笔位置的垂直距离。
- 注意：因为不是所有的字体都包含垂直度量，当FT_HAS_VERTICAL为假时，vertBearingX, vertBearingY和vertAdvance的值是不可靠的。

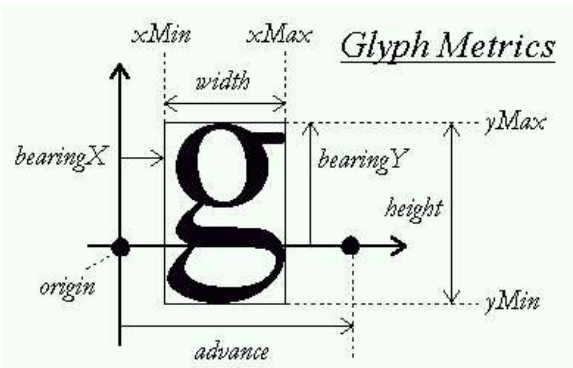
下面的图形更清楚地图解了度量。第一个图解了水平度量，其基线为水平轴：

android NDK编译多个sc	(19710)
C++移位运算符	(18351)
判断点在多边形内的多种	(17994)
GLSL 在OpenGL中向sh	(15022)
利用ffmpeg框架扩展and	(14858)
egl gles2.0开发	(13873)
android 音频介绍	(12259)
C++ memset详解	(12256)

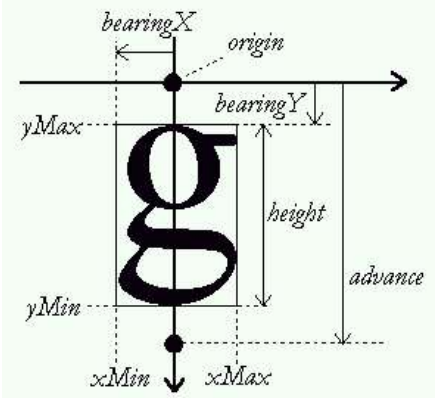
评论排行	
利用ffmpeg框架扩展and	(12)
Android IPC分析	(10)
GLSL 在OpenGL中向sh	(6)
判断点在多边形内的多种	(6)
android 使用icon进行字	(5)
android TabHost 例子	(4)
Android Audio audio_tra	(4)
Android 用SlidingDrawe	(4)
android NDK编译多个sc	(3)
Android 二级动画弹出菜	(3)

推荐文章	
*列式存储 HBase 系统架构学习	
*通过Android源码分析再探观察者模式(二)	
*浅析ZeroMQ工作原理及其特点	
*Rebound-Android的弹簧动画库	
*大型分布式网站术语分析	
*Hadoop中Map端shuffle源码解析	

最新评论	
Android RecyclerView 使用解析	明明爱: 也不写参考那大神的博客, 楼主这样不厚道啊...
C++移位运算符	棉花汤1990: 我用C++试了一下, j=1<i;j=1<<35;两个结果是一样的啊, 请教一下咋回事啊?
Android RecyclerView 使用解析	dasefg: Hi,最近在看Skia, 看得头大, 可以留个联系方式交流下吗
android 提升性能总结	丁国华: 不断的总结, 不断的进步。
android framework框架分析	51pansou: 谢谢! 想看 android 视频教程的可以到这个博客来 http://blog.csdn.net/r...
glBlendFunc函数	legendwin: glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA) 这...
opengl shader 绘制心形	hgl868: 着色器中 cos, sin, 等三角函数, 效率居然和角度大小有关, 角度值越大, 效率越低。
Skia深入分析	luoweifu: Skia是一个功能非常强大, 有代码庞大而复杂的一个开源库。目前也在研究, 一起学习。
游戏中的心理学	hus1259: 靠, 中枪无数。这是谁总结出来的。



对于垂直文本排列，基线是垂直的，与垂直轴一致：



Face->glyph->metrics中的度量通常以26.6像素格式（例如1/64像素）表示，除非你在调用FT_Load_Glyph或FT_Load_Char时使用了FT_LOAD_NO_SCALE标志，这样的话度量会用原始字体单位表示。

字形槽(glyph slot)对象也有一些其他有趣的字段可以减轻开发者的工作。你可以通过face->glyph->xxx访问它们，其中xxx是下面字段之一：

Advance
这个字段是一个FT_Vector，保存字形的经变换步长。当你通过FT_Set_Transform使用变换时，这是很有用的，这在第一部分的循环文本例子中已经展示过了。这个值是默认的(metrics.horiAdvance,0)，除非你在装载字形图像时指定FT_LOAD_VERTICAL，那么它将会为(0,metrics.vertAdvance)，这点与第一部分的例子不同。

linearHoriAdvance
这个字段包含字形水平推进宽度的线性刻度值。实际上，字形槽返回的metrics.horiAdvance值通常四舍五入为整数像素坐标（例如，它是64的倍数），字体驱动器用它装载字形图像。linearHoriAdvance是一个16.16固定浮点数，提供了以1/65536像素为单位的原始字形推进宽度的值。它可以用来完成伪设备无关文字排版。

linearVertAdvance
这与linearHoriAdvance类似，但它用于字形的垂直推进高度。只有当字体face包含垂直度量时这个值才是可靠的。

2.管理字形图像

装载到字形槽的字形可以转换到一幅字形位图中，这可以在装载时使用FT_LOAD_RENDER标志或者调用FT_Render_Glyph函数实现。每一次你装载一个新的字形到字形槽，前面装载的字形将会从字形槽中抹去。

但是，你可能需要从字形槽中提取这个字形，并在你的应用程序中缓存它，或者进行附加的变换，或者在转换成位图前测量它。

FreeType 2 API有一个特殊的扩展能够以一种灵活和普通的方式处理字形图像。要使用它，你首先需要包含FT_GLYPH_H头文件，如下：

```
#include FT_GLYPH_H
```

现在我们将解释如何使用这个文件定义的这个函数。

a.提取字形图像

你可以很简单地提取一个字形图像。这里有一向代码向你展示如何做：

