

FrontPanelText 示例

# *\*本示例适用于 Xbox One XDK（2017 年 6 月）*

# 描述

FrontPanelText 示例演示了如何使用 CPU 在 Xbox One X Devkit 前面板显示屏上绘制文本。本示例使用名为 RasterFont 的类，可加载 .rasterfont 文件。顾名思义，.rasterfont 文件包含一个简单的基于像素的字体，其中各字形都被栅格化。此格式适合在 CPU 上呈现。RasterFont 类提供了 printf 风格方法，让文本的呈现更为方便。有关如何使用电脑上安装的任何 True Type 字体自行创建 .rasterfont 文件的更多信息，另请参阅 RasterFontGen 示例。

# 使用样本

FrontPanelText 示例适用于带有集成式前面板的 Xbox One X DevKit。启动示例时，前面板显示屏上将会呈现一些示例文本。使用前面板方向键（左、右）更改文本的字体和字体大小（上、下）。使用方向键上按钮将放大字体，而使用方向键下按钮将缩小字体。



每个按钮对应 1 个 LED 灯

5X 可编程按钮

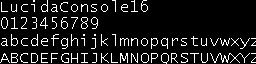
256 x 64 x 4bpp OLED 显示屏

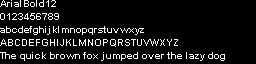
DPAD + 选择

也可以按下方向键按钮（选择）从前面板显示屏捕获缓冲区，并将结果保存到 Title Scratch 文件夹中的 .dds 文件。

下图是示例中的屏幕截图，显示了以不同大小呈现的几个字体选项：

在前面板上，Lucida Console 的可读性佳。这种字体宽度固定，因此是菜单和小组件的良好选择，因为文本的布局几何将可预测。此外，在 12 像素高度下仍然可读，且在这个尺寸下，显示屏可以轻松容纳 5 行文本。

这是以 16 像素大小呈现的相同字体。请注意，在此尺寸下，显示屏将可放置 4 行文本。

Arial 不是固定宽度，比较小写字母和大写字母的宽度便可看出。Arial 在 12 像素高度下可读，且与固定宽度的字体相比，可以在显示屏上水平放置更多文本。

使用 RasterFont 工具链，您可使用系统上安装的任何 True Type 字体生成 .rasterfont 文件。这是符号字体的示例。符号字体中的字符可用于呈现简单的 UI 元素（如箭头、按钮等）。

# 实施说明

使用 RasterFont 对象在 CPU 上呈现前面板的文本。如要创建 RasterFont 对象，请将 .rasterfont 文件的文件名传递给构造函数。例如：

auto myFont = RasterFont(L"Assets\\LucidaConsole16.rasterfont");

对于您自己的项目，您可使用示例随附的 .rasterfont 文件，也可使用 RasterFontGen.exe 工具创建自己的文件。RasterFontGen 可让您使用系统上安装的任何 True Type 字体创建具有各种大小和选项的 .rasterfont 文件。

以下代码片段显示了 RasterFont 对象的假设端到端使用情况：

// 加载 .rasterfont 文件

auto myFont = RasterFont(L"Assets\\LucidaConsole16.rasterfont");

// 获取前面板显示屏的缓冲区描述符

BufferDesc fpDesc = m\_frontPanelDisplay->GetBufferDescriptor();

// 将格式化字符串绘制到缓冲区

myFont.DrawStringFmt(fpDesc, 0, 0,

L"Simple Addition\n%i + %i = %i",

1, 1, (1 + 1));

// 将缓冲区在前面板呈现

m\_frontPanelDisplay->Present();

BufferDesc 是一种追踪 CPU 缓冲区宽度和高度的结构。RasterFont 可将文本呈现至内存中的任何地址，只需要一个描述缓冲区尺寸 BufferDesc。为更方便瞄定前面板显示屏，本示例使用了 FrontPanelDisplay 类以管理前面板缓冲区。使用 FrontPanelDisplay::GetBufferDescriptor() 获取适用于通过 RasterFont 将文本呈现至前面板的 BufferDesc。

DrawStringFmt 用于将文本绘制到缓冲区。这类似于标准库函数，printf()。请注意，它需要一个 BufferDesc 以及文本的 x 和 y 坐标。布局文本时，DrawStringFmt 支持换行符。

以下是 RasterFont 提供的文本呈现方法的简要描述：

// MeastureString 和 MeasureStringFMt 对计算很有用

// 用于布局用途的文本边界

RECT MeasureString(const wchar\_t \*text) const;

RECT MeasureStringFmt(const wchar\_t \*format, ...)const;

// 呈现到缓冲区的基本文本

void DrawString(const struct BufferDesc &destBuffer, unsigned x, unsigned y,

const wchar\_t \*text) const;

// 呈现到缓冲区的格式化文本

void DrawStringFmt(const struct BufferDesc &destBuffer, unsigned x, unsigned y,

const wchar\_t \*format, ...)const;

// 以下 DrawString 变型提供了阴影

// 可让您指定不同阴影的参数

// 灰色

void DrawString(const struct BufferDesc &destBuffer, unsigned x, unsigned y,

uint8\_t shade, const wchar\_t \*text) const;

void DrawStringFmt(const struct BufferDesc &destBuffer, unsigned x, unsigned y,

uint8\_t shade, const wchar\_t \*format, ...)const;

// 特定于字形的方法用于精确定位单个字形

RECT MeasureGlyph(wchar\_t wch) const;

void DrawGlyph(const struct BufferDesc &destBuffer, unsigned x, unsigned y,

wchar\_t wch, uint8\_t shade = 0xFF) const;

RasterFont 备注：

* 具有阴影参数的 DrawString() 变型可用于呈现白底黑色文本。例如，如果要在菜单系统中为选定行使用白底黑色文本。请记住， DrawString() 不会绘制背景像素，因此为获取白底黑色文本，首先必须绘制一个白色矩形。使用 MeasureString 方法确定所需矩形的边框。
* MeasureGlyph() 和 DrawGlyph() 对于精确定位各个字形非常有用。这些方法仅使用字形的边界框，不会考虑相邻字形之间的间距以及用于布置普通文本流的垂直偏移。这让您可精确定位字形。例如，如果您将符号字体中的字形用于 UI 元素或小组件。（您可参阅 FrontPanelDemo 示例了解一些 MeasureGlyph() 和 DrawGlyph() 的示例。）
* 如果您对 RasterFont 提供的基本文本流不满意，则可考虑使用基础 RasterGlyphSheet 类。该类提供了一个 ForEachGlyph() 模板，可用于编写自定义文本流实现。有关使用 ForEachGlyph() 的最佳方法，请参阅各种 RasterFont::DrawString\*() 方法的实现。

其他实施说明：

* 在 Xbox One X DevKit 上，::IsXboxFrontPanelAvailable() 将返回 true，而 ::GetDefaultXboxFrontPanel 将返回有效的 IXboxFrontPanelControl 对象。否则，::IsXboxFrontPanelAvailable() 将返回 false，而 ::GetDefaultXboxFrontPanel 将返回 nullptr（例如在 Xbox One、Xbox One S 或任何没有实体前面板的零售主机上。）
* 无需将每一帧都呈现在前面板上 (IXboxFrontPanelControl::PresentBuffer)。您只需要在一个或多个像素发生更改时呈现。因此，本示例有一个 m\_dirty 成员，只要显示缓冲区发生更改，就会设置该成员。
* 也只有在出现更改时才需要设置灯状态。
* 您无法直接访问前面板缓冲区。您必须管理自己的缓冲区并将缓冲区的地址传递至 IXboxFrontPanelControl::PresentBuffer。Sample::CaptureFrontPanelScreen() 只使用 m\_panelBuffer 的内容作为 DDS 表面的像素负载。

# 更新历史记录

2017 年 4 月，本示例首次发布。

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。