

FrontPanelDemo 示例

# *\*本示例适用于 Xbox One XDK（2017 年 6 月）*

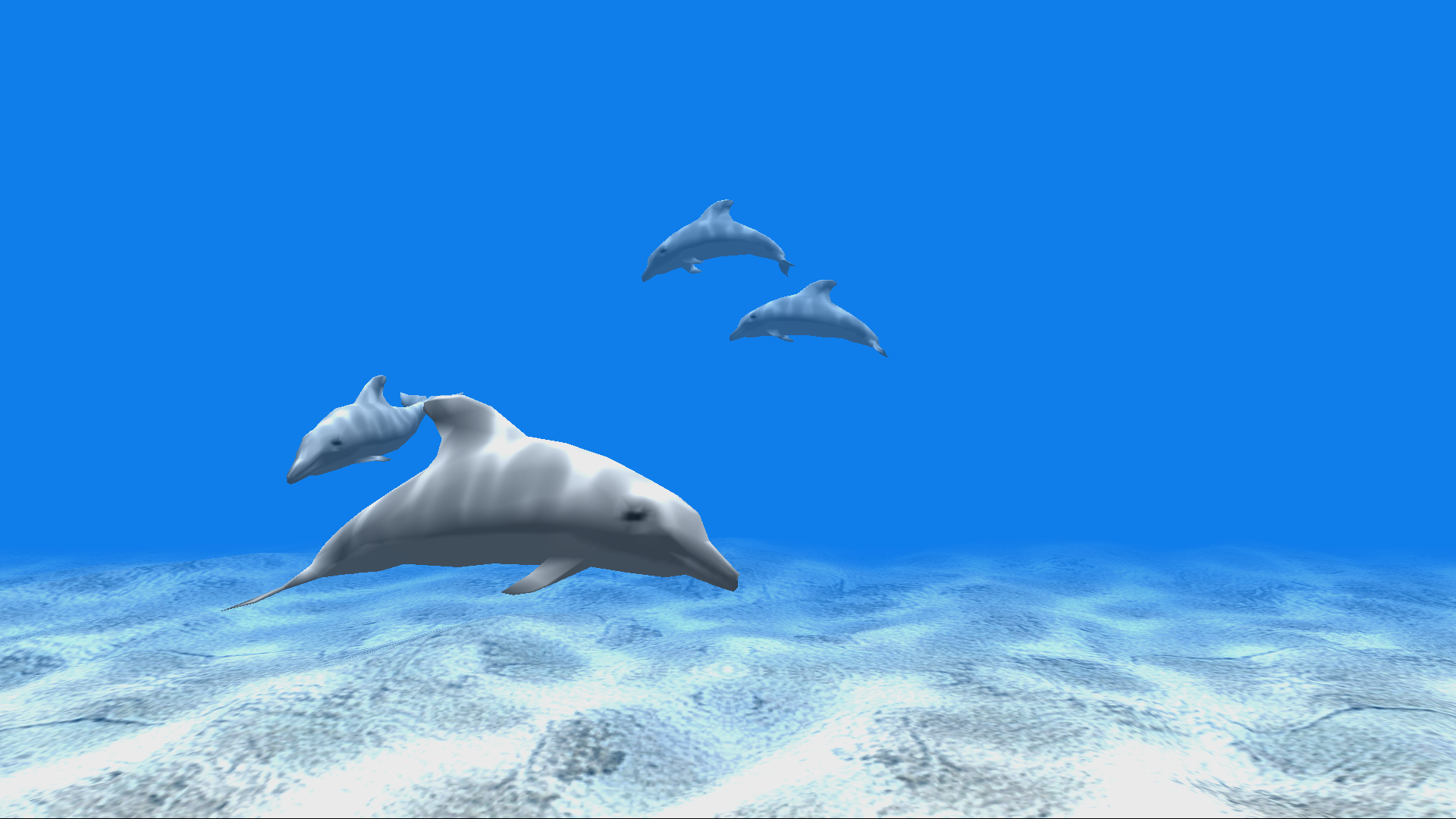
# 

# 描述

FrontPanelDemo 将多个示例组合为一个可执行文件，然后将功能与完全托管在 Xbox One X DevKit 前面板上的菜单系统进行绑定。顾名思义，其目的在于展示前面板的部分功能。有关此演示各部分的更详细说明，建议读者浏览其他前面板示例。

# 使用样本

FrontPanelDemo 示例在主显示屏上呈现简单的海豚场景，并在 Xbox One X DevKit 前面板上托管菜单驱动演示。请注意，FrontPanelDemo 适用于带有集成式前面板的 Xbox One X DevKit（该示例将在 Xbox One 或 Xbox One S 上编译和运行，但如果没有实体前面板，则可展示性不强。）



Xbox One X DevKit 前面板有 5 个切换按钮，5 个 LED 灯，一个 LCD 显示屏和一个 DPAD 输入设备。5 个 LED 灯与 5 个切换按钮分别进行了物理关联。LCD 显示屏宽 256 像素，高 64 像素，支持 16 级灰度。DPAD 支持上、下、左、右输入，也可向下按表示选择。



每个按钮对应 1 个 LED 灯

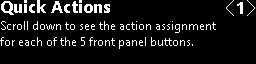
5X 可编程按钮

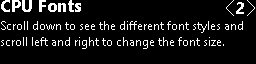
256 x 64 x 4bpp OLED 显示屏

DPAD + 选择

## 顶层菜单

菜单系统由四个顶层菜单组成，每个菜单都显示了您可从该菜单访问的功能的简单描述。使用 DPAD（向下）“向下滚动”并访问各菜单功能：

本示例具有多个操作，可动态映射到前面板上 5 个切换按钮中的任意一个。“快速操作”(Quick Actions) 菜单可用于检查各按钮的映射。

“CPU 字体”(CPU Fonts) 子菜单显示了在前面板显示屏上呈现的各种字体。有关呈现文本的更多信息，请参阅 FrontPanelText 示例。

“GPU 到前面板”(GPU to Front Panel) 演示了如何使用 GPU 渲染场景，然后在前面板上显示。有关更详尽示例

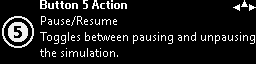
“按钮映射”(Button Mappings) 子菜单可用于将按钮动态重新映射到不同的操作。

## “快速操作”子菜单

从“快速操作”<1> 菜单中，使用方向键下按钮以转至“快速操作”子菜单。各子菜单说明了当前映射到前面板上相应切换按钮的操作。此外，当您导航到对应于切换按钮的子菜单时，与该按钮关联的 LED 会闪烁。

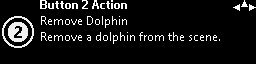
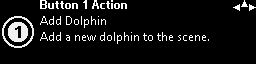
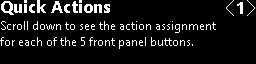
各子菜单还有一个字形，其上有一个数字用于指示相关的切换按钮。

请注意各子菜单右上角的“罗盘”。这为您提供了导航“提示”。在以上示例中，您可使用方向键左、方向键右和方向键上按钮。在整个演示过程中均依循此惯例。

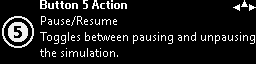


方向键导航

切换按钮



…



## CPU 字体

CPU 字体屏幕非常多，在此我们不一一展示。以下是一种字体的示例：

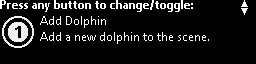
这显示了使用 RasterFontGen 工具以 32 像素高度生成的 Segoe UI 粗体。您可使用方向键左、右按钮以更改字体高度，使用方向键上、下按钮浏览不同字体。请注意，向上次数过多时，您最终会返回顶层菜单（CPU 字体 <2>）。有关如何使用 CPU 在 Xbox One X DevKit 前面板上呈现文本的更多详细信息，请参阅 RasterFontGen、RasterFontViewer 和 FrontPanelText 示例。

## GPU 到前面板

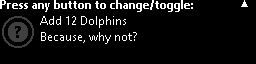
GPU 到前面板 <3> 有一个子菜单。这只是一个简单的屏幕，用于捕获主显示屏上呈现的内容，然后实时将其复制到前面板。 您可使用方向键上按钮返回顶层菜单（GPU 到前面板 <3>。）有关如何使用 GPU 渲染可前面板显示屏上使用的图形的更详细示例，请参阅 FrontPanelDolphin 示例。

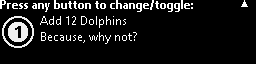
## 按钮映射

“按钮映射”(Button Mappings) 屏幕类似“快速操作”屏幕，但描述略有不同。各按钮映射屏幕对应于一个可映射到按钮的操作。可用的操作数超过了按钮数。因此，总会至少有一个操作未映射到按钮。因此，在“按钮映射”屏幕上，图标指示哪个按钮（如有）当前是该操作的“所有者”。



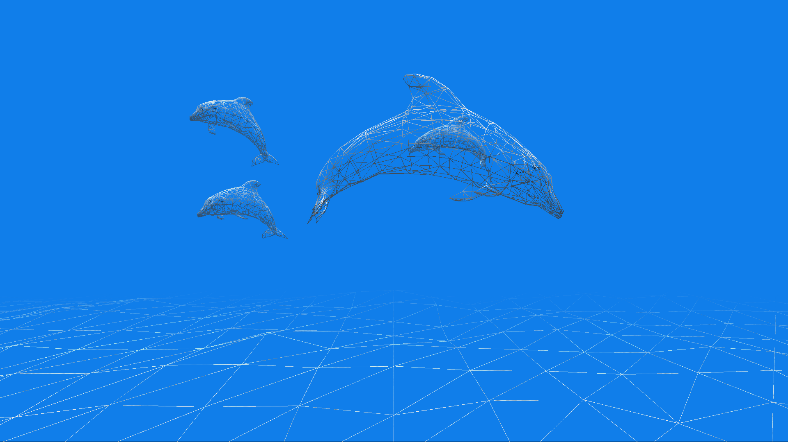
拥有切换按钮

在本示例中，按钮按钮图标变暗，且显示了一个问号，表示没有按钮拥有该操作。如果用户按下任何切换按钮，则该按钮将成为该操作的新所有者。此外，该按钮之前拥有的任何操作（如有）将不再拥有所有者。例如，按下按钮 1 后，相同的“添加 12 头海豚”(Add 12 Dolphins) 操作屏幕将会显示如下：



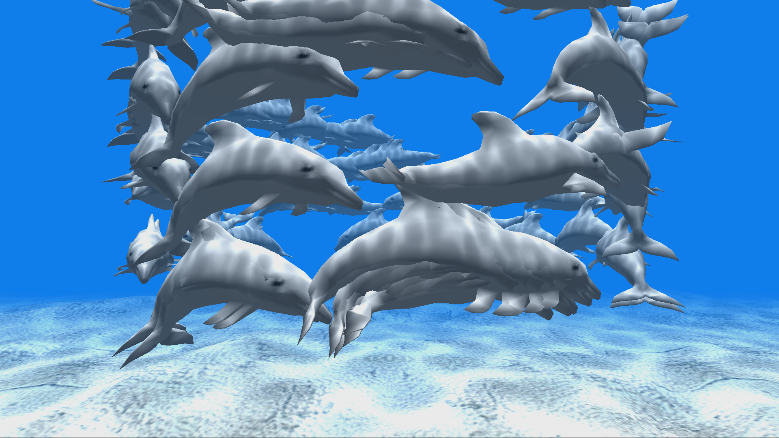
以下是按钮映射屏幕工作原理的汇总：

* 如果操作没有所有者，则按下任意按钮将使得该按钮成为该操作的新所有者。
* 如果操作已拥有所有者，则您可按下其他的切换按钮以更改所有者。
* 如果操作已拥有所有者，您按下了拥有切换按钮，则该操作将不再拥有所有者
* 如要实际执行操作，您必须使用方向键上按钮回滚到顶层。

本示例目前有 6 个可映射操作：

1. 添加海豚
2. 删除海豚
3. 清除所有海豚
4. 切换线框
5. 暂停/恢复模拟
6. 添加 12 头海豚

例如，左图显示了通过按下前面板按钮执行“切换线框”操作后以线框模式渲染的场景。

在这里，我们看到使用“添加 12 头海豚”操作添加许多海豚的结果。

# 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/)。