

FrontPanelDolphin 샘플

# *\*이 샘플은 Xbox One XDK와 호환됩니다 (2017 년 6 월)*

# 설명

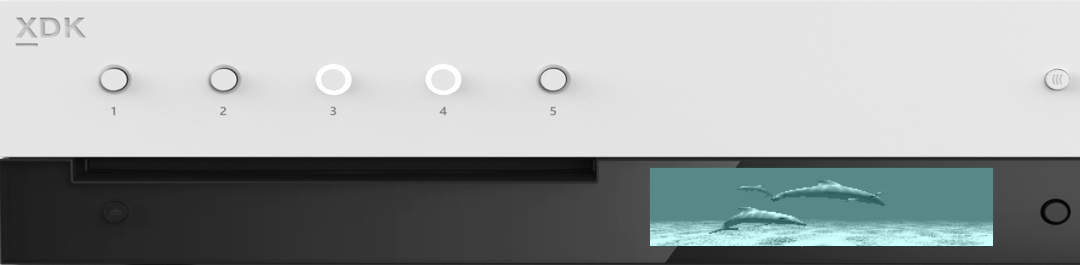
FrontPanelDolphin은 GPU를 사용하여 프FrontPanel에 렌더링하는 방법을 보여줍니다. 대부분의 게임 개발자는 GPU를 사용하여 화면에 렌더링할 수 있는 많은 코드를 보유 할 것으로 예상되므로 이 샘플의 목적은 기존 코드를 활용하여 프FrontPanel 디스플레이를 보다 쉽게 타겟팅할 수 있게 하는 것입니다.

다음은 몇 가지 사용 사례입니다.

* 실험실 환경에서 "헤드리스" devkit 을 사용하여 게임을 실행하고 프FrontPanel을 사용하여 일반적으로 화면에 표시되는 것을 렌더링하려고 합니다. 게임이 생신한 지는 한눈에 알 수 있습니다.
* 많은 게임 엔진에는 이미 리테일 버전에서 액세스할 수 없는 게임의 진단 기능을 사용할 수 있는 "개발 HUD"가 있습니다. 예를 들어, 비밀 컨트롤러 버튼 조합을 수행하면 HUD가 표시됩니다. 그런 다음 HUD는 테스트 및 개발 중 유용한 몬스터 생성, 특정 레벨로 점프 또는 캐릭터를 무적으로 만들기와 같은 추가 옵션을 제공합니다. 이 HUD는 일반적으로 GPU에서 렌더링되므로, 결과를 프FrontPanel에 복사할 수 있다면 기존 HUD 코드를 재사용하여 프런트 패널에 적용할 수 있습니다. 프FrontPanel에 배치하면 화면의 실제 공간을 희생하지 않아도 됩니다. 또한 게임 패드를 사용하는 대신 DPAD 및 프FrontPanel의 버튼을 사용할 수 있습니다.

# 샘플 사용하기

FrontPanelDolphin 샘플은 통합된 프FrontPanel이 있는 Xbox One X DevKit 용입니다. 샘플을 실행하면 돌핀 장면이 주 디스플레이에 표시되고 동시에 프FrontPanel LCD 디스플레이에 렌더링됩니다.



버튼당 1개의 LED

5x 프로그래밍 가능한 버튼

256 x 64 x 4bpp OLED 디스플레이

DPAD + 선택

## FrontPanelDolphin 주 디스플레이

## FrontPanelDolphin 프FrontPanel 디스플레이



샘플은 프FrontPanel의 DPAD 선택 버튼을 제외한 모든 입력을 처리하지 않습니다. 선택 버튼을 누르면 프FrontPanel 디스플레이에서 버퍼를 캡처하고 결과를 타이틀 스크래치 폴더에 있는 .dds 파일에 저장합니다.

# 구현 정보

이 샘플에서는 FrontPanelRenderTarget라고 부르는 도우미 클래스를 사용합니다. 이름에서 알 수 있듯이 프런트 패널 디스플레이에 적합한 오프 스크린 렌더 타겟입니다. FrontPanelRenderTarget 제공된 텍스처를 회색조로 변환한 다음 쿼드로 렌더링합니다. 이는 매우 간단한 버텍스 셰이더와 매우 간단한 픽셀 셰이더를 사용하여 수행됩니다. 버텍스 셰이더는 쿼드를 생성하고 픽셀 셰이더는 제공된 텍스처를 샘플링하고 내적을 사용하여 각 픽셀을 그레이 스케일로 변환합니다. 렌더링 단계는GPUBlit ()라고 부르는 메서드로 구현됩니다 :

// 제공된 쉐이더 리소스 뷰를 소스로 사용하여 그레이 스케일 이미지를 렌더링합니다.

void GPUBlit(ID3D11DeviceContext \*context, ID3D11ShaderResourceView \*srcSRV);

제공된 ID3D11ShaderResource 보기는 렌더링 할 텍스처에서 만들어야 합니다.

GPUBlit ()을 호출한 후 결과를 CPU의 버퍼로 다시 복사한 다음 버퍼를 프FrontPanel 디스플레이에 제공해야 합니다. 그만큼 FrontPanelRendertTarget 클래스는 이 작업을 쉽게 수행할 수 있는 두 가지 방법을 제공합니다.

// 렌더링 대상을 스테이징 텍스처에 복사 한 다음 CPU에 다시 복사합니다.

void CopyToBuffer(ID3D11Device \*device,

ID3D11DeviceContext \*context, ATG::BufferDesc &desc);

// 렌더링 타겟을 스테이징 텍스처에 복사하고 결과를 CPU에 복사합니다.

// 그런 다음 글꼴 패널 표시에 표시합니다.

// 호출 할 때마다 여분의 버퍼를 할당합니다.

void PresentToFrontPanel(ID3D11Device \*device, ID3D11DeviceContext \*context);

BufferDesc는 CPU 버퍼의 너비와 높이를 추적하는 구조체입니다. FrontPanelRenderTarget::CopyToBuffer 메모리에 있는 모든 주소로 복사할 수 있습니다. 필요한 것은 버퍼 크기를 기술한 BufferDesc입니다. 샘플에서는 프론트패널용 BufferDesc 을 얻기 위한 프런트 패널용 버퍼 관리와 FrontPanelDisplay::GetBufferDescriptor() 를 사용하는 FrontPanelDisplay 클래스를 사용합니다. 그런 다음 FrontPanelRenderTarget에서 FrontPanelRenderTarget::CopyToBuffer() 이미지를 불어와야 합니다. 마지막으로 프FrontPanel 디스플레이에 이미지를 정확히 실제로 표시하기 위해서 FrontPanelDisplay::Present() 를 불러와야 합니다.

FrontPanelRenderTarget::PresentToFrontPanel() 방법은 CPU에 복사하고 버퍼에 두는 두 가지 과정을 관리함을 알아두시기 바랍니다. FrontPanelDisplay 클래스를 아직 사용하지 않는 경우에 편리합니다. FrontPanelRenderTarget::PresentToFrontPanel()의 단점은 불러올 때마다 버퍼를 할당해야 한다는 것입니다.

프FrontPanel 렌더링을 위해 돌핀 샘플을 프FrontPanel에 적용하는 것은 FrontPanelRenderTarget을 사용해서 실제로는 매우 간단했습니다. 몇가지 변경만 하면 최소한 적응할 수 있습니다:

**Sample::Sample:**

// 프런트 패널 렌더링 타겟을 구축

m\_frontPanelRenderTarget = std::make\_unique <FrontPanelRenderTarget>();

// 기본 프런트 패널 가져 오기

DX::ThrowIfFailed(GetDefaultXboxFrontPanel(m\_frontPanelControl.ReleaseAndGetAddressOf()));

// FrontPanelDisplay 객체를 초기화합니다.

m\_frontPanelDisplay = std::make\_unique<FrontPanelDisplay>(m\_frontPanelControl.Get());

**Sample::CreateDeviceDependentResources:**

// GPU를 통해 프런트 패널에 렌더링하기 위한 설정

ComPtr<IXboxFrontPanelControl> frontPanelControl;

DX::ThrowIfFailed(GetDefaultXboxFrontPanel(frontPanelControl.ReleaseAndGetAddressOf()));

// 프FrontPanel 렌더 타겟 리소스를 생성

m\_frontPanelRenderTarget->CreateDeviceDependentResources(frontPanelControl.Get(),

device);

**Sample::CreateWindowSizeDependentResources:**

// 메인 렌더 타겟에 대한 쉐이더 리소스 뷰를 생성

auto device = m\_deviceResources->GetD3DDevice();

DX::ThrowIfFailed(device->CreateShaderResourceView(m\_deviceResources->GetRenderTarget(),

nullptr, m\_mainRenderTargetSRV.GetAddressOf ()));

**Sample::Render:**

// 프FrontPanel 렌더 타겟에 블리트한 다음 프FrontPanel에 표시

m\_frontPanelRenderTarget->GPUBlit(m\_deviceResources->GetD3DDeviceContext(),

m\_mainRenderTargetSRV.Get());

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

m\_frontPanelRenderTarget->CopyToBuffer (m\_deviceResources->GetD3DDevice (),

m\_deviceResources->GetD3DDeviceContext (), fpDesc);

m\_frontPanelDisplay->Present();

# 업데이트 기록

2017 년 4 월 샘플의 첫 번째 릴리스.

# 개인정보처리방침

샘플을 컴파일하고 실행할 때 샘플 실행 파일의 이름이 Microsoft로 보내져 샘플 사용을 추적 할 수 있습니다. 이 데이터 수집을 거부하려면 Main.cpp에서 "샘플 사용 텔레메트리"라는 코드 블록을 제거하면 됩니다.

Microsoft의 개인 정보 취급 방침에 대한 일반적인 내용은 [Microsoft 개인 정보 취급 방침](https://privacy.microsoft.com/ko-kr/privacystatement/)을 참조하십시오.