LABORATÓRIO 4

COMUNICAÇÃO CPU-GPU

EXERCÍCIOS DE APRENDIZAGEM

FAÇA OS EXERCÍCIOS PARA FIXAR O CONTEÚDO

1. Modifique o **projeto MBuffers_DXUT** de forma que seja possível visualizar que alocador está sendo usado para armazenar os comandos de desenho e acompanhar como as cercas são usadas para fazer a sincronização entre CPU e GPU. Uma saída de exemplo é mostrada abaixo.

```
Init
Fence: 0
Allocator [1]: Draw
Fence: 1
Allocator [0]: Draw
Fence: 2
Allocator [1]: Draw
Fence: 3
Allocator [0]: Exit
```

Para isso modifique o método **SyncToGpu** da classe Graphics, inserindo mensagens de depuração que serão exibidas na aba Output do Visual Studio ao rodar o projeto em modo Debug:

```
void Graphics::SyncToGpu()
{
	// insere uma nova cerca na fila de comandos da GPU
	const ullong currentFenceValue = fenceValue[bufferIndex];
	ThrowIfFailed(commandQueue->Signal(fence, currentFenceValue));

#ifdef _DEBUG
	TCHAR msg[20];
	wsprintf(msg, "Fence: %d\n", currentFenceValue);
	OutputDebugString(msg);
#endif

	// troca de buffer
	bufferIndex = (bufferIndex + 1) % bufferCount;
```

```
#ifdef _DEBUG
	wsprintf(msg, "Allocator [%d]: ", bufferIndex);
	OutputDebugString(msg);
#endif

	// espera a GPU completar os comandos do buffer
	if (fence->GetCompletedValue() < fenceValue[bufferIndex])
	{
		// aciona evento quando a GPU atingir a cerca
		ThrowIfFailed(fence->SetEventOnCompletion(
		    fenceValue[bufferIndex],
		    fenceEvent));

		// espera até o evento ser acionado
		WaitForSingleObject(fenceEvent, INFINITE);
	}

	// ajusta o valor da cerca para o próximo frame
	fenceValue[bufferIndex] = currentFenceValue + 1;
}
```

Modifique também os métodos **Init**, **Draw** e **Finalize** da classe MBuffers para que eles exibam mensagens. O objetivo dessas mudanças é simular como será a operação da nossa aplicação gráfica. Quando a aplicação estiver completa, o código de inicialização e desenho irão inserir comandos na lista de comandos. Esses comandos precisarão ser enviados para a GPU com uma chamada a SendToGpu:

```
void MBuffers::Init()
{
    OutputDebugString("Init\n");

    // envia comandos de inicialização
    graphics->SendToGpu();
}

void MBuffers::Draw()
{
    OutputDebugString("Draw\n");

    // envia comandos de desenho
    graphics->SendToGpu();
}

void MBuffers::Finalize()
{
    OutputDebugString("Exit\n");
}
```