

UNIVERSIDADE PRESBITERIANA MACKENZIE
FACULDADE DE COMPUTAÇÃO E INFORMÁTICA - FCI
COMPUTAÇÃO VISUAL - 07G

Aluno: Daniel Faia Monteiro da Silva

TIA: 31932029

Pesquisa sobre APIs gráficas

- **O que é uma API?**

O termo **API** significa **Application Programming Interface**, ou Interface de Programação de Aplicativos, em português. As APIs são um conjunto de ferramentas utilizadas para construir os aplicativos que utilizamos e também para fazer a comunicação entre suas partes e diferentes componentes.

Uma boa analogia às APIs são os garçons de um restaurante. O **garçom** ouve o **cliente**, anota seus **pedidos** e leva-os à cozinha. Quando os pedidos estão prontos, o garçom retorna com os pratos e **entrega** ao cliente.

Nos dispositivos, a API funciona de forma parecida: ela recebe algum pedido do sistema operacional ou aplicativo e o leva até o destino, fazendo a tradução da mensagem para uma linguagem que o receptor compreenda.

Depois de trabalhada, a informação é novamente entregue à API, que fica encarregada de retornar até a origem e traduzi-la para a forma adequada. Além disso, são as APIs que permitem a comunicação entre diferentes aplicativos, sistemas operacionais, peças de hardware e até mesmo entre aparelhos distintos.

- **O que é uma API gráfica?**

As APIs gráficas são uma variante dedicada especificamente à área de imagem e processamento gráfico 3D, fazendo todo o trabalho de comunicação entre os jogos, drivers e a placa de vídeo em si. É por causa delas que jogadores do mundo inteiro podem optar por marcas como NVIDIA, AMD ou Intel, vendo seus jogos renderizados corretamente.

As APIs gráficas ficam em uma camada entre as engines (os motores gráficos) dos jogos e o driver. O driver é o responsável pela tradução final, levando ao hardware instruções precisas do que precisa ser feito e entregue de volta aos aplicativos.

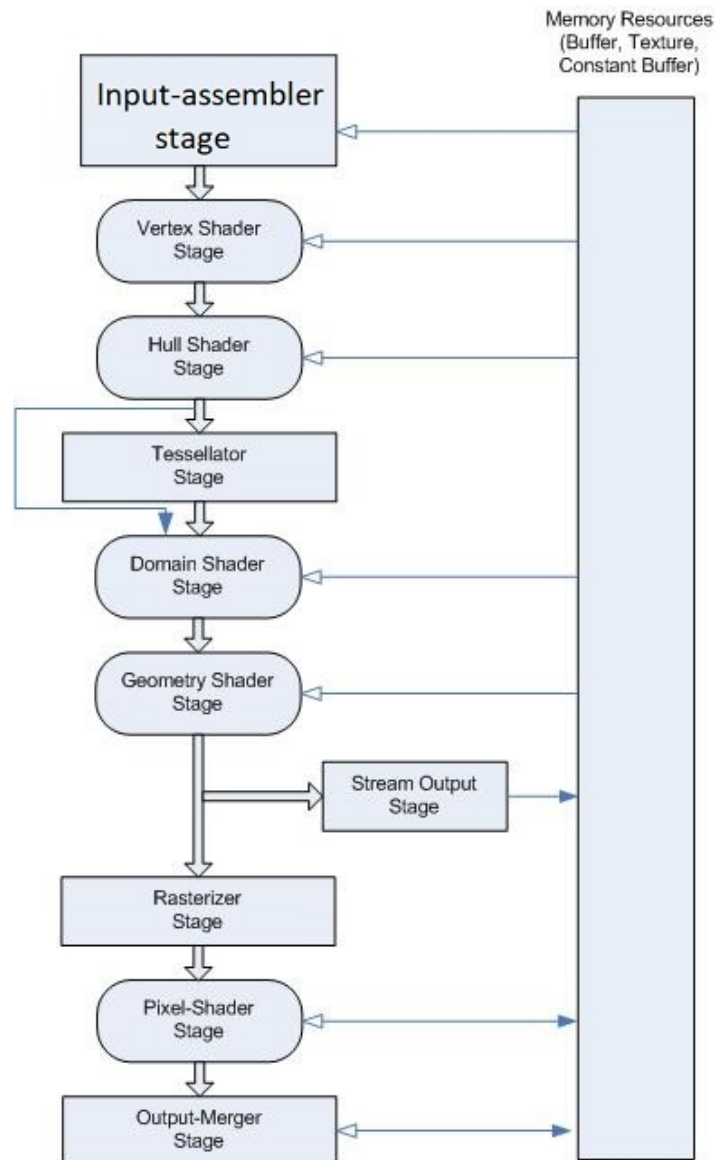
- **DirectX e Vulkan**

Atualmente, as APIs gráficas mais famosas e mais utilizadas no mercado são a DirectX, que pertence à empresa Microsoft, e a Vulkan, que pertence às empresas Khronos e AMD.

Essas APIs são chamadas de low level devido à sua proximidade com o hardware (camada física). Além disso, elas permitem que os programadores elaborem códigos mais próximos do hardware quando necessário, o que resulta em maior desempenho.

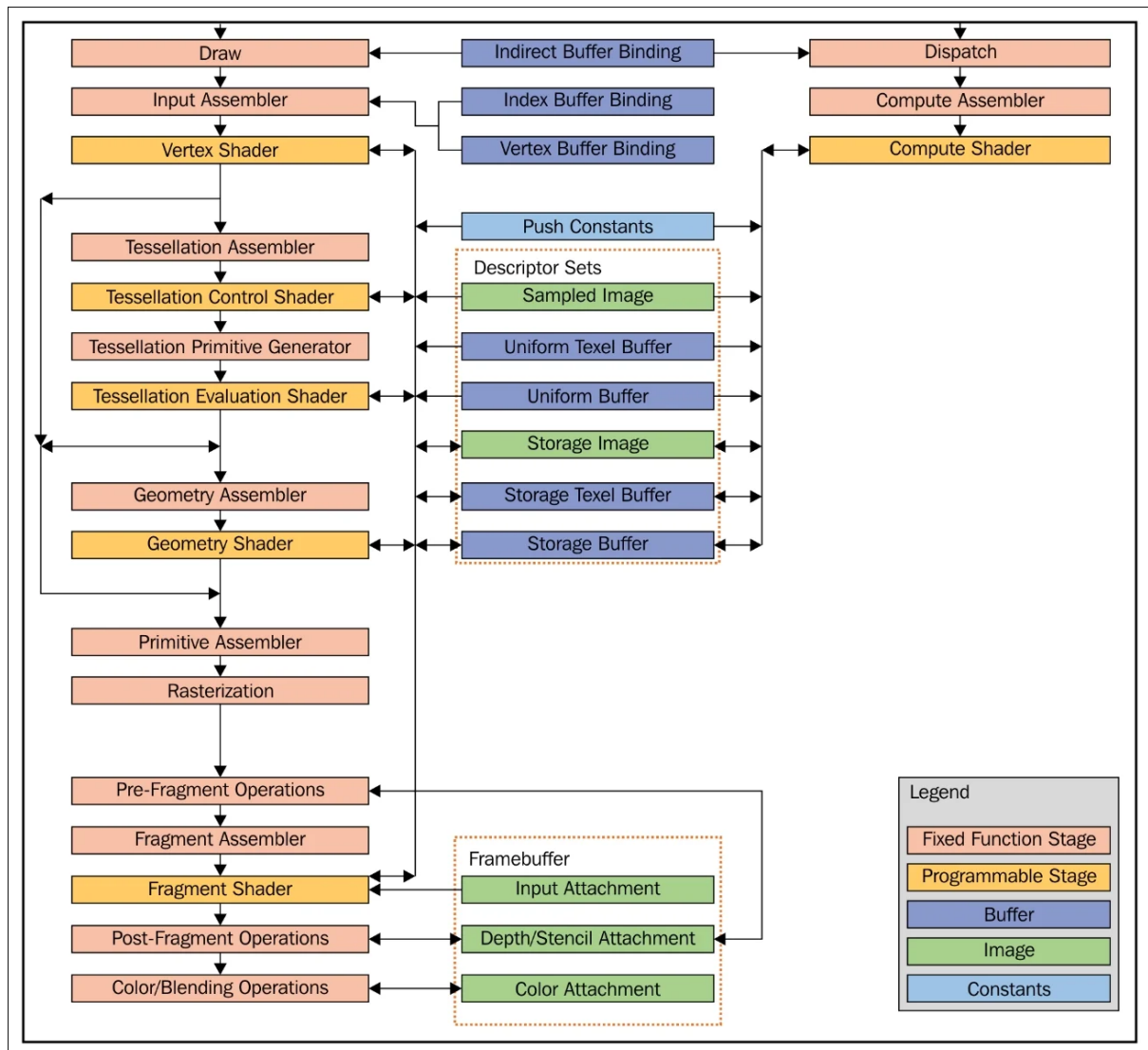
- **Pipeline e Shading DirectX**

De acordo com a empresa Microsoft, o pipeline do DirectX é documentado como mostra a imagem abaixo. Além disso, a linguagem de shading utilizada é uma linguagem chamada HLSL (High-level shader language).



- Pipeline e Shading Vulkan

De acordo com a empresa Khronos, o pipeline do Vulkan é documentado como mostra a imagem abaixo. A princípio, a API Vulkan só fornecia suporte a linguagem de shading GLSL (**OpenGL Shading Language**). Entretanto, atualmente também existe suporte à linguagem HLSL citada anteriormente.



- “Hello World”, Pixel Shader e aplicações do DirectX

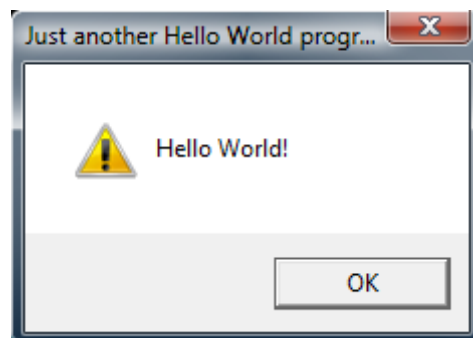
Um exemplo de código para imprimir um “Hello World” gráfico utilizando o DirectX é mostrado na imagem abaixo:

```
#include <windows.h>    // include the basic windows header file

// the entry point for any Windows program
int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine,
                  int nShowCmd)
{
    // create a "Hello World" message box using MessageBox()
    MessageBox(NULL,
               L"Hello World!",
               L"Just another Hello World program!",
               MB_ICONEXCLAMATION | MB_OK);

    // return 0 to Windows
    return 0;
}
```

A saída desse código seria uma janela imprimindo o “Hello World”, como mostra a imagem abaixo:

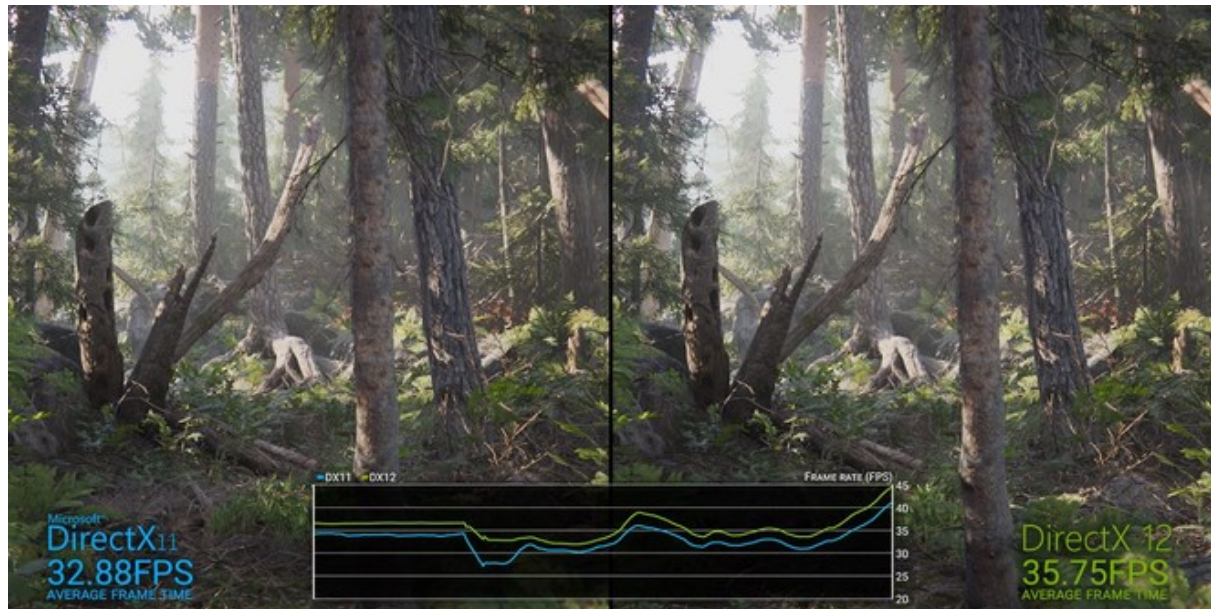


Abaixo um exemplo de código Shader do DirectX:

```
struct PixelShaderInput{
    float4 position : SV_POSITION;
    float3 outVec : POSITION0;
    float3 normal : NORMAL0;
    float3 light : POSITION1;
};

float4 main(PixelShaderInput input) : SV_TARGET{
    float3 L = normalize(input.light);
    float3 V = normalize(input.outVec);
    float3 R = normalize(reflect(L, input.normal));
    float4 diffuse = Ka + (lightColor * Kd * max(dot(input.normal, L), 0.0f));
    diffuse = saturate(diffuse);
    float4 specular = Ks * pow(max(dot(R, V), 0.0f), shininess.x - 50.0f);
    specular = saturate(specular);
    float4 finalColor = diffuse + specular;
    return finalColor;
}
```

Como citado anteriormente, o DirectX é uma das APIs mais utilizadas no mercado de games. Por ser da Microsoft e estar diretamente relacionado com a NVIDIA, a maior fabricante de placas de vídeo do mundo, pode-se dizer que o DirectX está presente em praticamente todos os jogos famosos do mercado e até mesmo em alguns não tão famosos assim. Portanto, o melhor exemplo de aplicação da API do DirectX são os jogos.



- **Aplicações do Vulkan**

Os exemplos de código da API Vulkan são muitos extensos e portanto não são propícios para se colocar neste documento pois ocuparia muito espaço, mas neste link do GitHub, feito por Sascha Willems, há vários exemplos de código utilizando o Vulkan. Segue o link: [SaschaWillems/Vulkan: Examples and demos for the new Vulkan API \(github.com\)](https://github.com/SaschaWillems/Vulkan).

Assim como o DirectX, o Vulkan também tem seu foco voltado ao mundo dos games e portanto, suas aplicações são bem parecidas com a API concorrente. Entretanto, como foi desenvolvido pela AMD, que não vende tantas placas de vídeo como a NVIDIA, ele acaba não sendo tão utilizado como o DirectX, apesar de exercer a mesma função.

REFERÊNCIAS

Blog 2 AM Gaming. **O QUE É API GRÁFICA E O QUE ISSO INFLUENCIA NO DESEMPENHO DOS SEUS JOGOS?** Setembro, 2019. Disponível em: <[O que é API gráfica e o que isso influencia no desempenho dos seus jogos? | 2 A.M. Gaming \(2amgaming.com\)](#)>. Acesso em: 08 Nov 2022.

Blog 2 AM Gaming. **O QUE É VULKAN E POR QUE OS JOGOS PRECISAM DISSO?** Dezembro, 2019. Disponível em: <[O que é Vulkan e por que os jogos precisam disso? | 2 A.M. Gaming \(2amgaming.com\)](#)>. Acesso em: 08 Nov 2022.

Microsoft. **Pipeline de elementos gráficos.** Setembro, 2022. Disponível em: <[Pipeline de elementos gráficos - Win32 apps | Microsoft Learn](#)>. Acesso em: 08 Nov 2022.

Microsoft. **High-level shader language (HLSL).** Abril, 2021. Disponível em: <[High-level shader language \(HLSL\) - Win32 apps | Microsoft Learn](#)>. Acesso em: 08 Nov 2022.

NGUYEN, Hay. **HLSL as a First Class Vulkan Shading Language.** Khronos Blog. Janeiro, 2020. Disponível em: <[Khronos Blog - The Khronos Group Inc](#)>. Acesso em: 08 Nov 2022.

Wikipedia. **GLSL.** Março, 2021. Disponível em: <[GLSL – Wikipédia, a enciclopédia livre \(wikipedia.org\)](#)>. Acesso em: 08 Nov 2022.

WILLEMS, Sascha. **Vulkan.** 2019. Disponível em: <[SaschaWillems/Vulkan: Examples and demos for the new Vulkan API \(github.com\)](#)>. Acesso em: 08 Nov 2022.