

# Intro to DirectX

---

Jens-Stefan Mikson

# What is DirectX?

Collection of APIs for handling tasks related to multimedia, especially game programming and video on Microsoft platforms.

- Direct3D - Graphics API. Rendering 3D with performance in mind.
- Direct2D - Vector graphics.
- DirectDraw - 2D graphics.
- DirectPlay - Network communication for game development. (deprecated)
- DirectSound - Interface to sound card drivers (superseded by XAudio2)
- DirectInput - Collect input from input devices (not supported)

DirectX is an umbrella term for the above.

Xbox - Console based on DirectX technology.

# Non-deprecated APIs in DirectX

- Direct3D - real time 3D rendering.
- DXGI - enumerates adapters and monitors. Manages swap chain.
- Direct2D - 2D graphics API.
- DirectWrite - Text rendering API.
- DirectCompute - general-purpose computing on GPUs (parallelizing work).
- DxDiag - diagnosing and generating reports related to DirectX components.
- XACT3 - High-level audio API. (Xbox only).
- XAudio2 - Low-level audio API. Successor of DirectSound.
- DirectX Raytracing (DXR) - real time raytracing.
- DirectStorage - GPU-oriented file I/O API.
- DirectML - GPU-accelerated machine learning and artificial intelligence API.

# Version history

DirectX 1.0 - Released in 1995 as Windows Game SDK.

...

DirectX 9 - Released in 2002. Introduced Shader Model 2.0.

DirectX 10 - Released in 2006.

DirectX 11 - Released in 2009. Shader Model 5.0.

DirectX 12 - Released in 2015.

# Direct3D

- Graphics API
- Uses hardware acceleration if available on the GPU. Accelerates the whole rendering pipeline or a part of it.
- Supports:

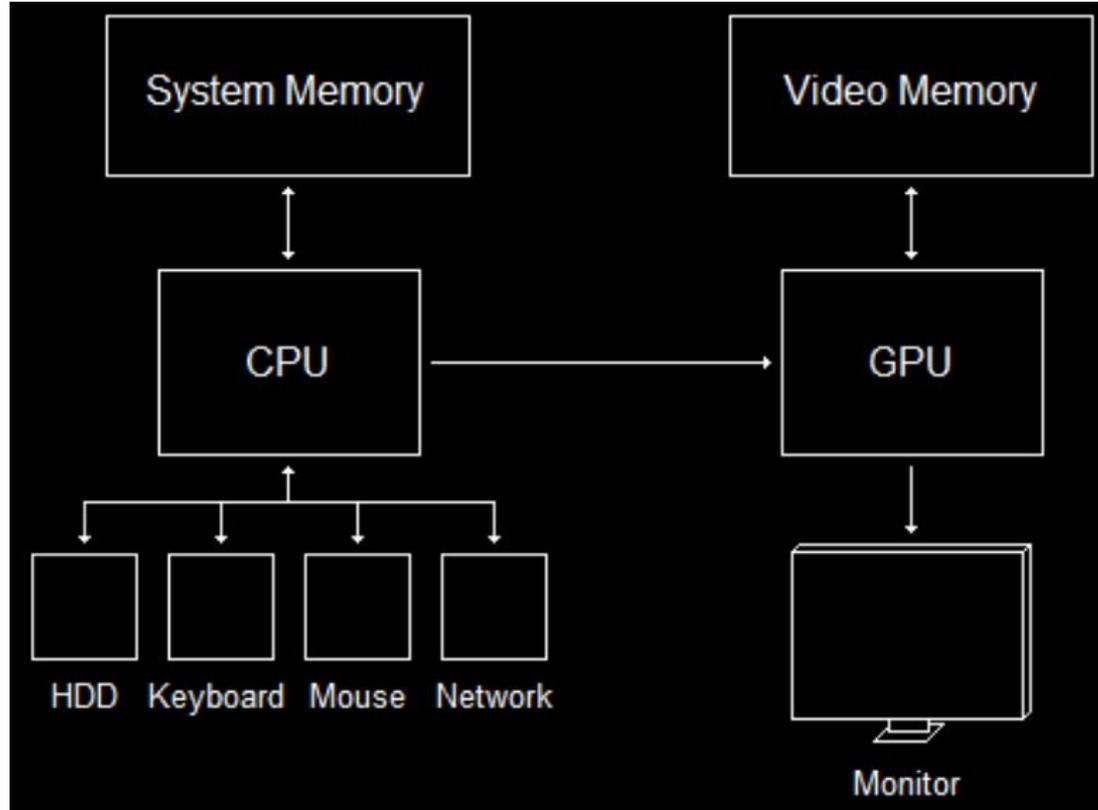
mipmapping  
texture blending  
atmospheric effects  
stencil buffering  
color blending  
culling  
Z-buffering  
W-buffering  
clipping  
alpha blending  
spatial anti-aliasing  
programmable HLSL  
shaders and effects  
perspective-correct texture mapping  
2D graphics

# Direct3D 11 vs Direct3D 12

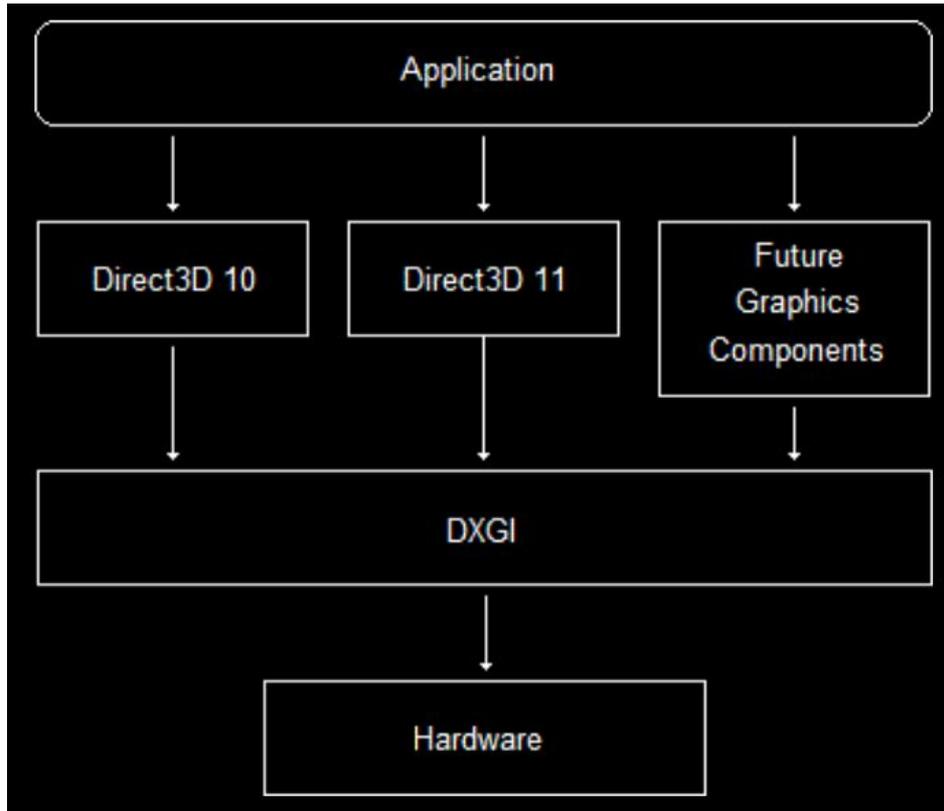
- Both support Shader Model 5.1.
  - Direct3D 12 allows a lower level of hardware abstraction.
  - Improved multithreaded scaling and improved CPU utilization.
- 
- Direct3D 12 is much more difficult to learn. Start with Direct3D 11.

Let's look at some theory for working with DirectX 11.

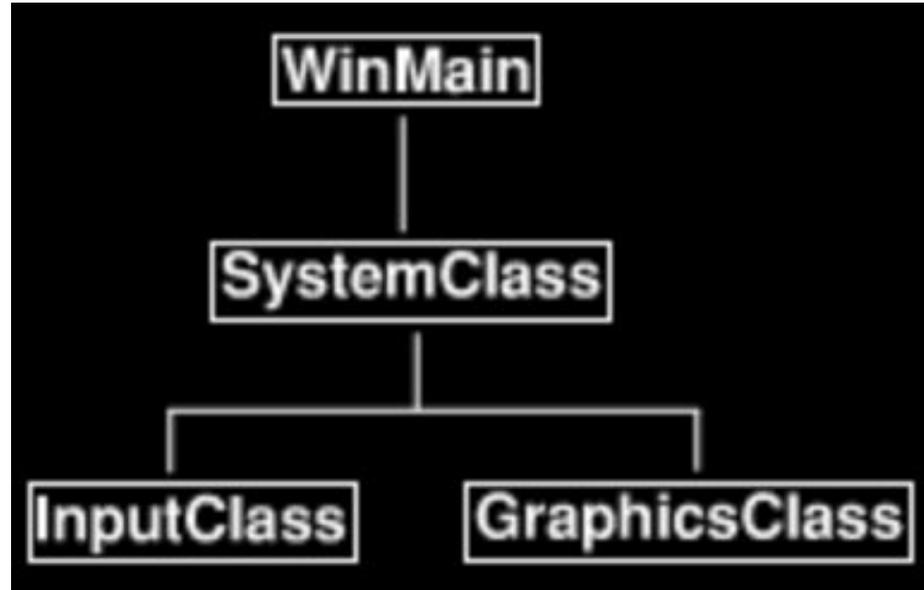
# Graphics Hardware



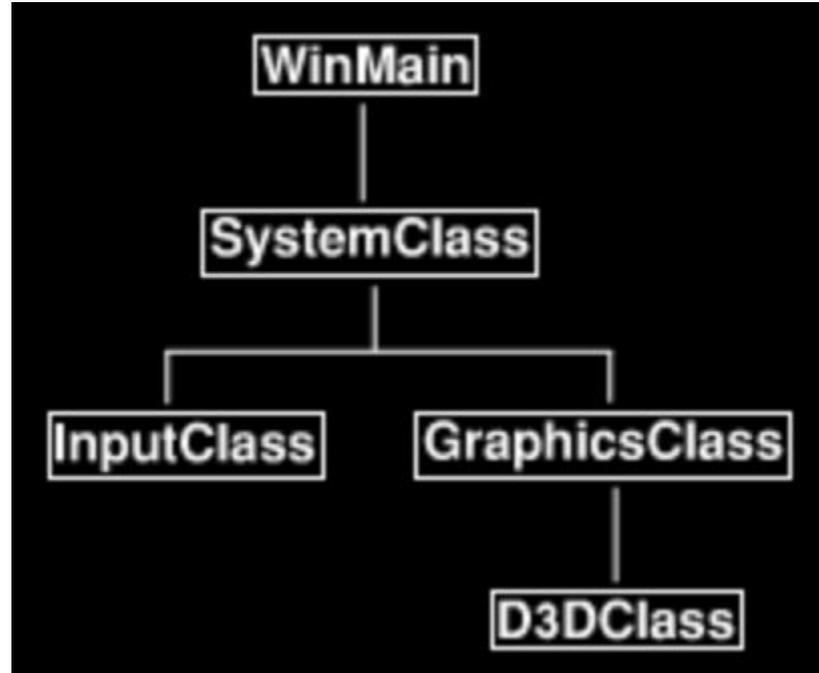
# DirectX Graphics Infrastructure (DXGI)



# Framework 1

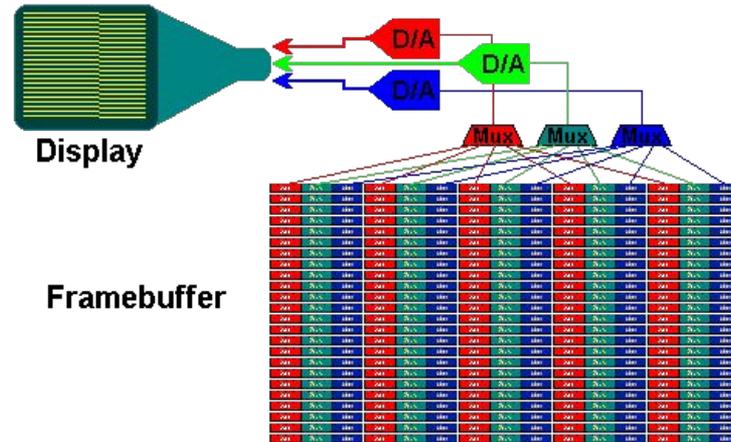


## Framework 2



# Image Buffer

- In GPU memory there is a pointer to a buffer of pixels, which form the image, which is displayed on the screen.
- When you want to draw a new image on the monitor, you update this buffer of pixels.



# Slow monitors

Monitors do not refresh as needed for real-time rendering.



# Tearing



# Tearing - Why?

1. Monitor is in the process of drawing the image.
2. While the monitor is drawing, we are rendering into our image buffer.
3. Monitor still keeps on drawing but draws the new values.
4. You get an image on the screen with pixels from the past buffer and the new buffer.

# Swapping

Use 2 buffers: front buffer and back buffer.

Render images to the back buffer. When done, DXGI updates the front buffer, discarding the old image.

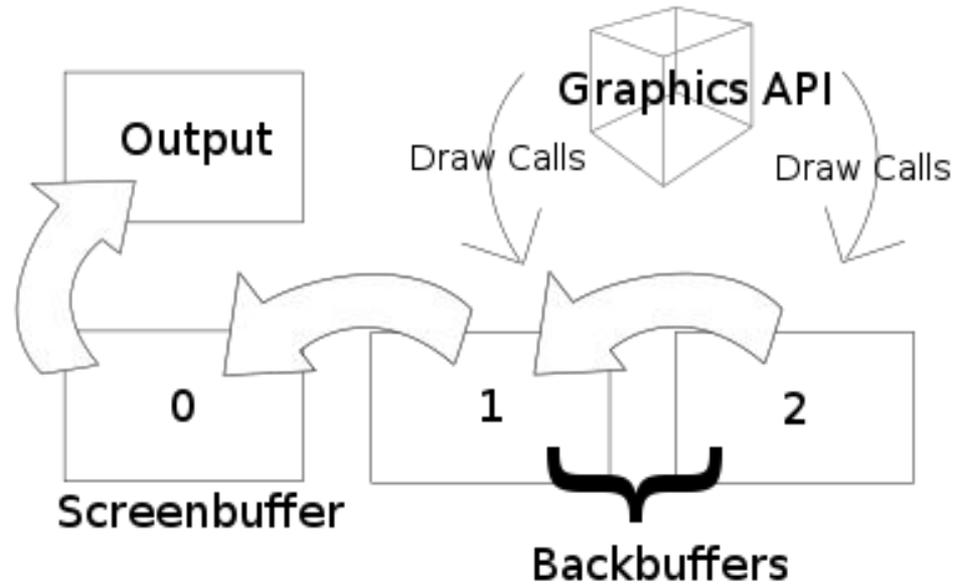
This happens via changing pointers to the buffers, i.e, very fast.

Is there a problem here?

# Swapping - Solution

The GPU asks the monitor when it's about to refresh. The GPU can time the swapping of the buffer to synchronize with the refresh rate of the monitor.

# The Swap Chain



## Framework 3

