

# Overview of 3D APIs for Graphics Cards

RNDr. Róbert Bohdal, PhD.

# 3D APIs for Graphics Cards

API = Application Programming Interface

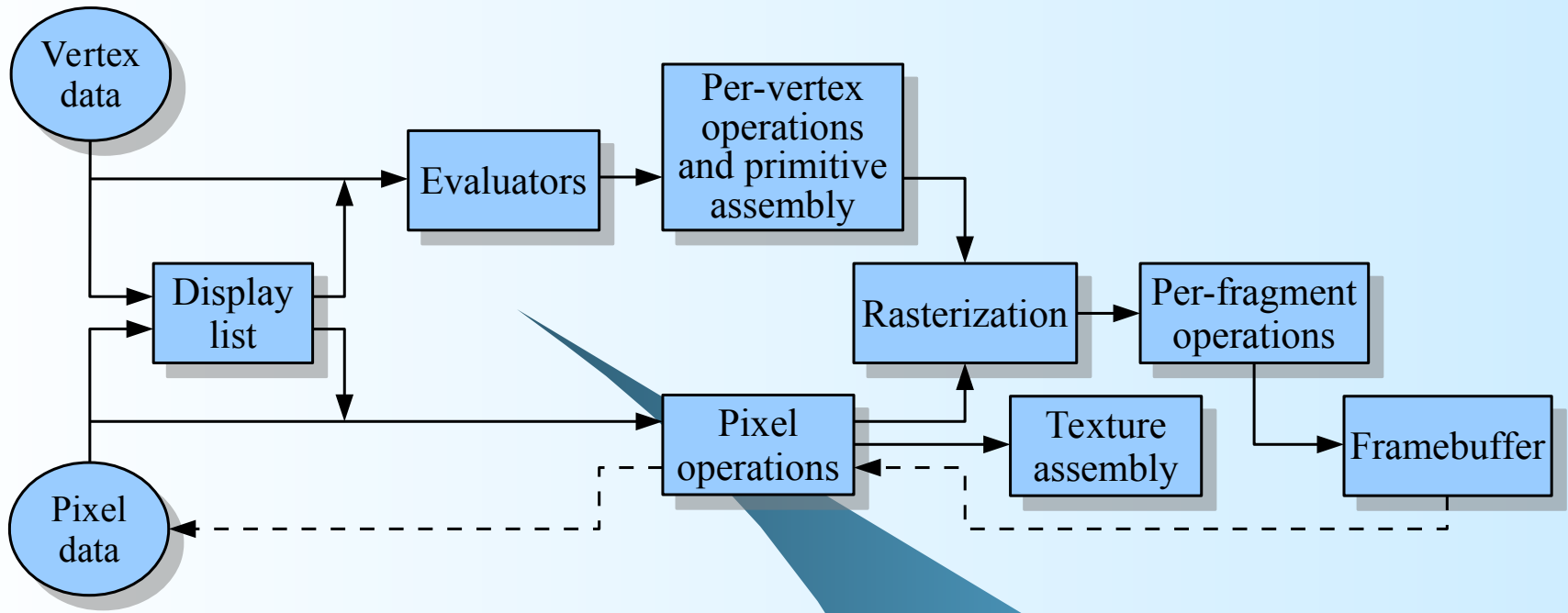
- OpenGL (Open Graphics Library)
- Direct3D (part of DirectX)
- Farenheit – was an effort to unify Direct3D and OpenGL. The project worked shortly, from 1997 to 2000.
- SDL (Simple DirectMedia Layer)
- Vulkan – low-overhead, cross-platform 3D graphics and computing API.



# History of OpenGL

- 1990' – *Silicon Graphics Inc.* (SGI) started using *IRIS GL* on its graphics workstations. Opening this specification was not possible, so SGI created OpenGL from the previous version of IRIS GL.
- 1992 – The *OpenGL ARB* (3Dlabs, Apple, ATI, Dell, IBM, Intel, Microsoft, Nvidia, SGI, and Sun) consortium approved OpenGL 1.0 as a standard.
- In 1993, Microsoft left the project.
- 2004 – OpenGL 2.0 has been released. It includes support for GLSL (OpenGL Shading Language).
- 2008 – OpenGL 3.0 (texture arrays, geometry shader, instancing, conditional rendering, ...)
- 2010 – OpenGL 4.0 (tessellation, OpenGL ES, texture compression, shader packing, ...)
- 2017 – the current version is OpenGL 4.6

# Pipeline of OpenGL



Pipeline – The sequence of steps that are performed when displaying input data, whether *vertex data* (vertices, lines, polygons), or *pixel data* (pixels, textures, bitmaps).

# Pipeline of OpenGL

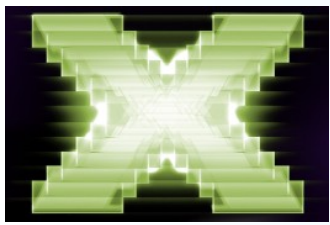
- **Display List:** The data can be stored for later use into the display list. In so-called *immediate* mode, data are processed instantly.
- **Evaluators:** Geometric data are entered using vertices. Curves and surfaces are usually given by control points. Evaluators calculate individual points (with normal and interpolated color) on the surface.
- **Per-vertex operations:** Vertices are given by 3D coordinates and transformed by a  $4 \times 4$  matrix to determine their position on the screen. If textures are used, texture coordinates are first calculated and then transformed. When lighting is turned on, the color value of the vertices is calculated with respect to the material and location of the lights.

# Pipeline of OpenGL...

- **Primitive Assembly:** In this section, the input primitives are clipped with respect to the viewing frustum/box and the invisible polygons are removed.
- **Pixel Operations:** Pixels data are “unpacked” and converted to the required format. They are then scaled, transformed and optionally stored or further processed in the rasterization block.
- **Texture Assembly:** Here the individual textures are mapped on the polygons of geometric objects.
- **Rasterization:** Rasterization is the conversion of vertex and pixel data into fragments. Each square fragment corresponds to a specific pixel in the framebuffer.

# Pipeline of OpenGL...

- **Fragment Operations:** Before the data are stored in the framebuffer, multiple operations are performed. The first is usually “texturing”, where for each fragment is assigned texture from the texture memory. After that, the *fog* may also be applied. Next, *scissor*, *alpha*, *stencil* and *depth-buffer test* is used. Then, *blending*, *dithering*, *logic operations* on pixels are performed and the *mask* is finally applied.



# History of DirectX

- 1994 – By introducing *Windows 95*, it became necessary to design an API that could effectively communicate with the graphics card.
- 1995 – Microsoft released the first version called *Windows Games SDK*.
- 2004 – The latest version 9.0c, which does not need *Windows Vista*, was released. (Xbox 360, pixel shader 2.0, ...).
- 2006 – DirectX 10 was released (only for OS Windows Vista and greater) (extensive changes in pipeline, unified shader model, windows display driver model, cube map arrays, ...).
- 2009 – DirectX 11 (GPGPU support, tessellation, stereoscopic 3D, DirectXMath, ...).
- 2012 – current version is DirectX 12.0 (Xbox One, DirectX raytracing, depth bounds testing, programmable multisample anti-aliasing, multi-GPU support, ...).



# Parts of DirectX

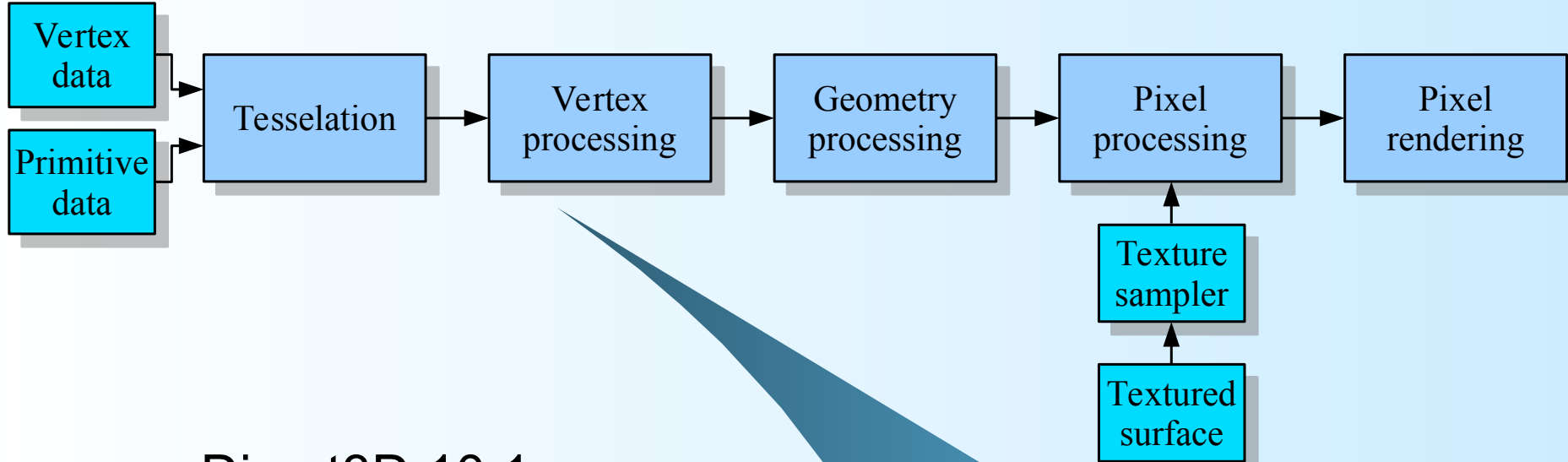
- **DirectX Graphics:**
  - *DirectDraw* – Allows direct access to the frame-buffer of the graphics card.
  - *Direct3D* (D3D) – draws 3D graphics primitives. It has 2 modes – *retained* and *immediate*.
- **DirectInput:** processes data from joystick, mouse, keyboard, etc.
- **DirectSound:** allows play and record sound.
- **DirectPlay:** allows connect *multiplayer* games over the network (LAN, modem) with others.
- **DirectMusic:** used to play songs that are created with *DirectMusic Producer*.

# Parts of DirectX...

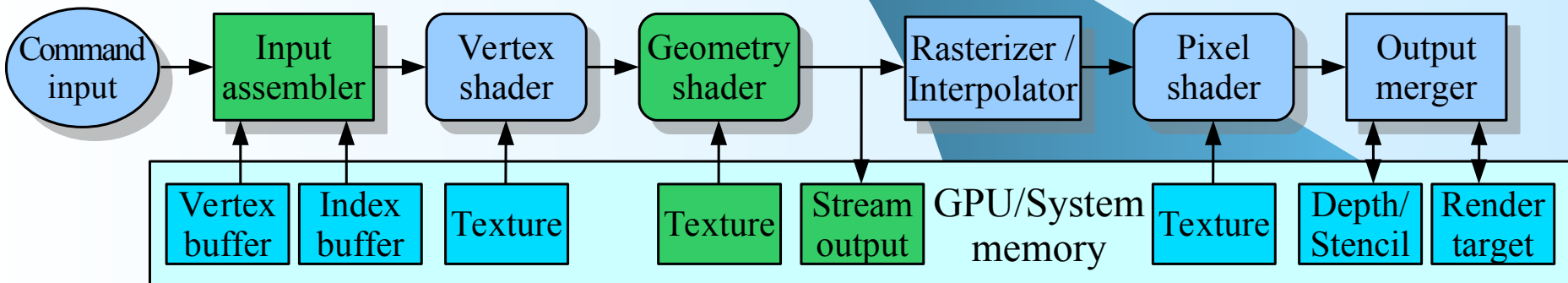
- **DirectX Media:** contains from *DirectShow*, *DirectAnimation* and *DirectX Transform*. These are mainly used in *media streaming* applications.
- **DirectX Media Objects:** provides support for *streaming objects* such as decoders and encoders.

# Pipeline of Direct3D 9.0 and 10.1

## Direct3D 9.0



## Direct3D 10.1



# Pipeline of Direct3D 10.1

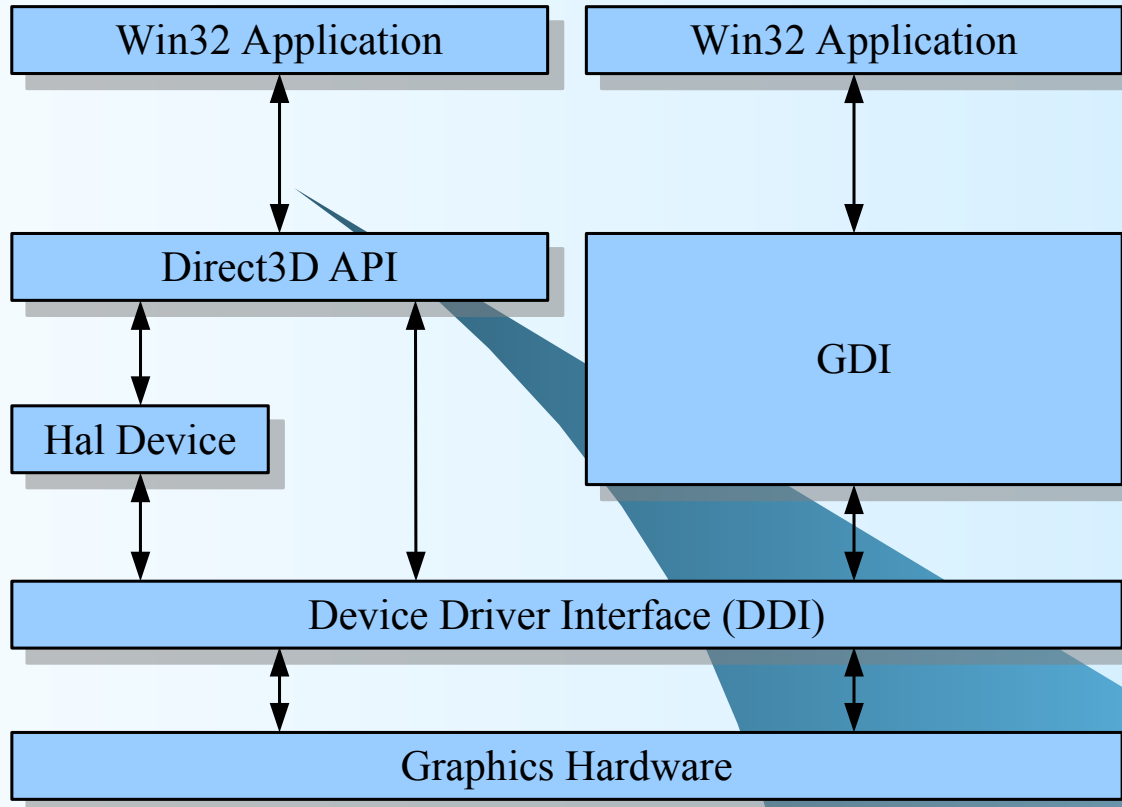
It consists of several programmable or functionally fixed blocks. Memory objects (buffers, textures) can be created in one block and then assigned to others for further processing.

- **Input assembler:** reads vertices and converts its to a floating point representation, assigns them a unique *id*.
- **Vertex shader:** is a programmable block that transforms vertices. It can be used for custom views, animation, diffuse lighting, etc.
- **Geometry shader:** is the block in which the individual primitives are processed. It reads vertices, lines, triangles, and creates one or more output primitives.

# Pipeline of Direct3D 10.1...

- **Stream output:** is a functionally fixed block that writes the vertex data into the memory.
- **Rasterization:** It performs a number of functions, including cropping, computation of triangle elements (equation of triangle plane, normal), and rasterization of individual primitives into sequences of pixels.
- **Pixel shader:** It is a block in which individual pixels are processed and the color calculation is performed.
- **Output merger:** Performs a *depth* and *stencil* test along with *blending* operations. It is a unique block because it needs not only to write data into memory but also to read data from memory.

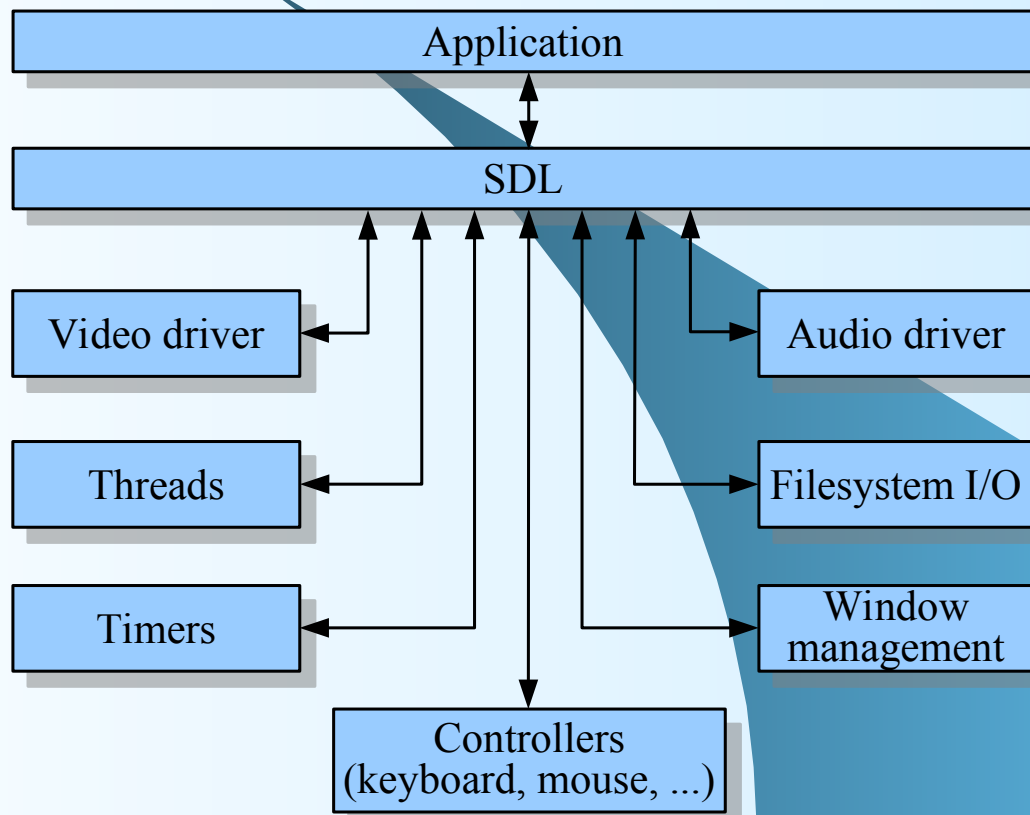
# Using Direct3D API in Windows





# Simple DirectMedia Layer

- It is a platform-independent multimedia library that provides low-level access to the sound system, keyboard, mouse, joystick and 3D hardware via OpenGL or videobuffer. It is used especially for programming simple games.





- Vulkan was first announced by the non-profit *Khronos Group* in 2015.
- Targets high-performance realtime 3D graphics applications across all OS (Windows, linux, ...) and platforms (PC, mobile, ...).
- Offers lower overhead, more direct control over the GPU, and lower CPU usage.
- Memory management tasks are left to the application and allows disabling of excessive validation and shader pre-compilation.
- It is considerably lower level API than OpenGL and offers parallel tasking.
- Has HLSL support and ray tracing for Nvidia graphics cards.
- Future plans include automatic multi-GPU features and OpenCL merging.



# Vulkan

