

# Development of Games

Lecture 12

Introduction to DirectX (Cont.)

# Structure of DirectX

- DirectX is a multimedia API that provides a standard interface to interact with graphics and sound cards, input devices and more
- For development of game it is needed to install DirectX SDK
- The nine DLL files roughly correspond to the ten namespaces in DirectX

# List of DirectX 9.0 namespaces

Namespace	Description
Microsoft.DirectX	Common Classes and math Structures
Microsoft.DirectX.Direct3D	3D graphics and helper libraries
Microsoft.DirectX.DirectDraw	Direct Draw graphics API. This is a legacy namespace and you should not need to use it.
Microsoft.DirectX.DirectPlay	Networking API for multiplayer games
Microsoft.DirectX.DirectSound	Sound support
Microsoft.DirectX.DirectInput	Input device support (i.e mouse and joystick)
Microsoft.DirectX.AudioVideoPlayback	Play Video and Audio (i.e playback a DVD on your PC)
Microsoft.DirectX.Diagnostics	Troubleshooting
Microsoft.DirectX.Security	Access security
Microsoft.DirectX.Security.Permissions	Access security permissions

Andrey V. Gavrillov

Kyung Hee University

# DirectDraw Concepts

# Setting Up DirectDraw

- DirectDrawCreate() or CoCreateInstance()
- Get the *IID\_IDirectDraw* interface pointer
- SetCooperativeLevel()
- SetDisplayMode()
- Fill in a DDSURFACEDESC
- CreateSurface()

# Flipping Surfaces

- "Front surface" and "Back surface"
- Create a "flipping chain"
- For double-buffering, the primary surface is created with one "attached surface" -- this is called the back buffer
- `GetAttachedSurface()`
- Draw on the back buffer
- Call `Flip()` on the primary

# Notes on Flipping

- `lpddsBack` is always the same and `lpddsFront` is always the same
- A `Flip()` does not flip these pointers as far as you're concerned
- You're simply drawing on different RAM

# Render of a Surface

- `IDirectDrawSurface::Lock()`
- Party on the pointer
- `IDirectDrawSurface::Unlock()`

# “Where’s PutPixel()?”

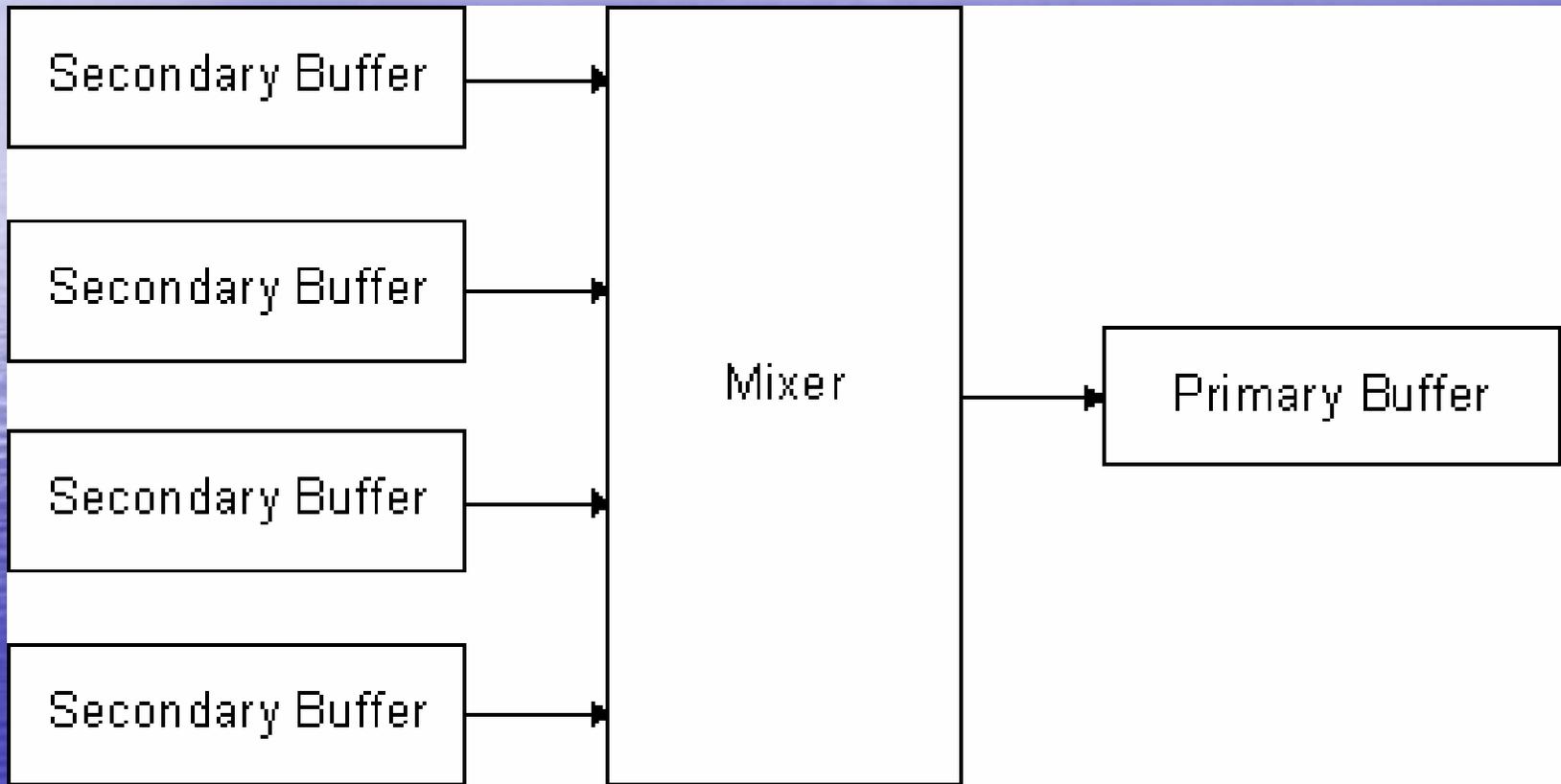
- DirectDraw is low-level
- There are no such drawing routines
- GDI, but it’s **butt slow**
- Write your own
- Most algorithms are easily adapted to DirectDraw if you simply account for the Lock, Unlock, and the IPitch

# DirectSound Concepts

# The DSound Object Model

- DirectSoundCreate()
- DirectSound object abstracts a sound card
- DirectSoundBuffer objects each represent a single audio source
  - Primary buffers
  - Secondary buffers

# DirectSound Buffers



# Setting Up DirectSound

- DirectSoundCreate() or CoCreateInstance()
  - CLSID\_DirectSound, IID\_IDirectSound
  - Initialize()
- SetCooperativeLevel()
- Fill Out a DSBUFFERDESC structure
- Call IDirectSound::CreateSoundBuffer()
- Party Away