FFmpeg

mpv

VLC

Hardware Acce

Lossy Compres

Lossless Comp

OpenGL

Direct3D

Multimedia, Graphics & Hardware

Hitesh Kumar Saini

56195

 @alexmercerind

 @alexmercerind

Course Instructor: Dr. Subodh Prasad
Department of Information Technology, College of Technology, G. B. P. U. A. & T., Pantnagar.

Introduction:
Video: Pixel, frame & representations.

- Every video file consists of individual frames.

- Every video frame consists of individual pixels.

- Every video pixel consists of 3 channels: Red, Green & Blue (RGB).
  - Not necessary!
    - Other pixel formats exist: RGB, RGBA, BGRA, ARGB, YUV, NV12 etc. etc.
    - Generally only relevant for in-memory (CPU RAM) representation.

A video pixel!

- Each pixel essentially represents a color (using RGB channel).
- Web-developers may already understand:
  - Familiar representations:
    - `rgb(120, 80, 35)`
    - `rgba(120, 80, 35, 255)`
    - `#121212`
    - `#121212ff`
- Each channel (R, G & B) has a range of 0 to 255.
- Can be represented/stored in 8 bits i.e.
  - $2^8 = 256$.
- Thus, memory taken by each RGB pixel $= 8 * 3$ bits $= 24$ bits $= 3$ bytes
- Thus, memory taken by each RGBA pixel $= 8 * 4$ bits $= 32$ bits $= 4$ bytes
- Thus, memory taken by RGB FHD 1080p frame $= 3 * 1920 * 1080$ bytes
- Thus, memory taken by RGBA FHD 1080p frame $= 4 * 1920 * 1080$ bytes

Introduction:
Video: Pixel, frame & representations.

- Size of a **25 FPS FHD** video of **1 second** in bytes:
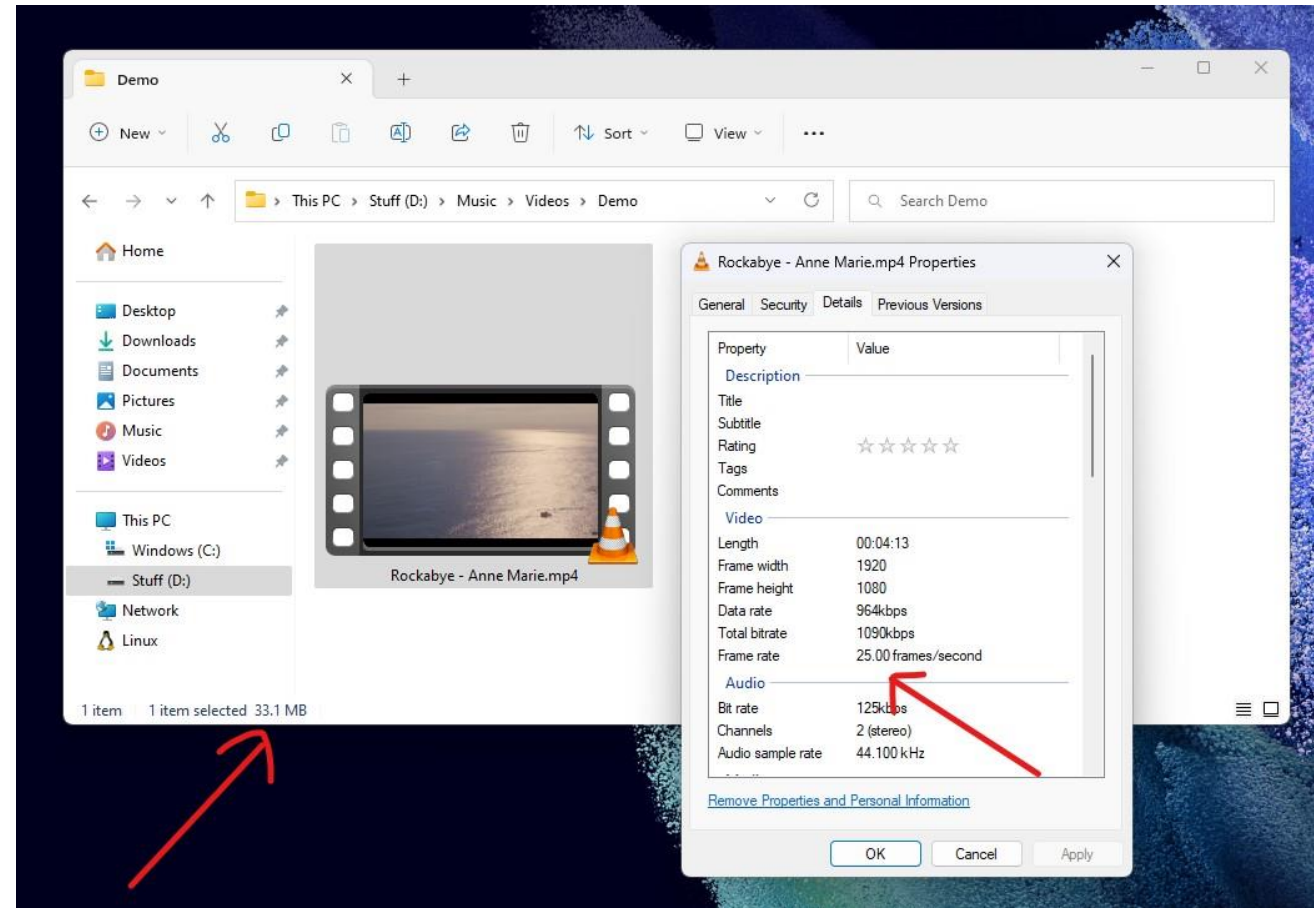
    = 25 * 4 * 1920 * 1080 bytes

    = 207,360,000 bytes

    = 207,360,000 / (1024 * 1024) MB

    = 197.75 MB!

- For comparison, here's a **25 FPS FHD** video of **4:13 minute (253 seconds)** consuming only 33.10 MB. You might already be familiar with size of video files downloaded from the internet.

- What makes it possible?
    - Compression Algorithms.
    - File Information:
        - Video Stream: H.264
        - Audio Stream: AAC

Introduction:
An example problem from "Data Communications & Networking" by Behrouz A. Forouzan.

Example 3.20

What is the bit rate for high-definition TV (HDTV)?

Solution

HDTV uses digital signals to broadcast high quality video signals. The HDTV Screen is normally a ratio of 16 : 9 (in contrast to 4 : 3 for regular TV), which means the screen is wider. There are 1920 by 1080 pixels per screen, and the screen is renewed 30 times per second. Twenty-four bits represents one color pixel. We can calculate the bit rate as

$$1920 \times 1080 \times 30 \times 24 = 1{,}492{,}992{,}000 \text{ or } 1.5 \text{ Gbps}$$

Questions worth consideration:
1. Why multiplied by 24?
2. What makes it possible transmitting (network) & displaying (locally) at this extremely high rate & magnitude.
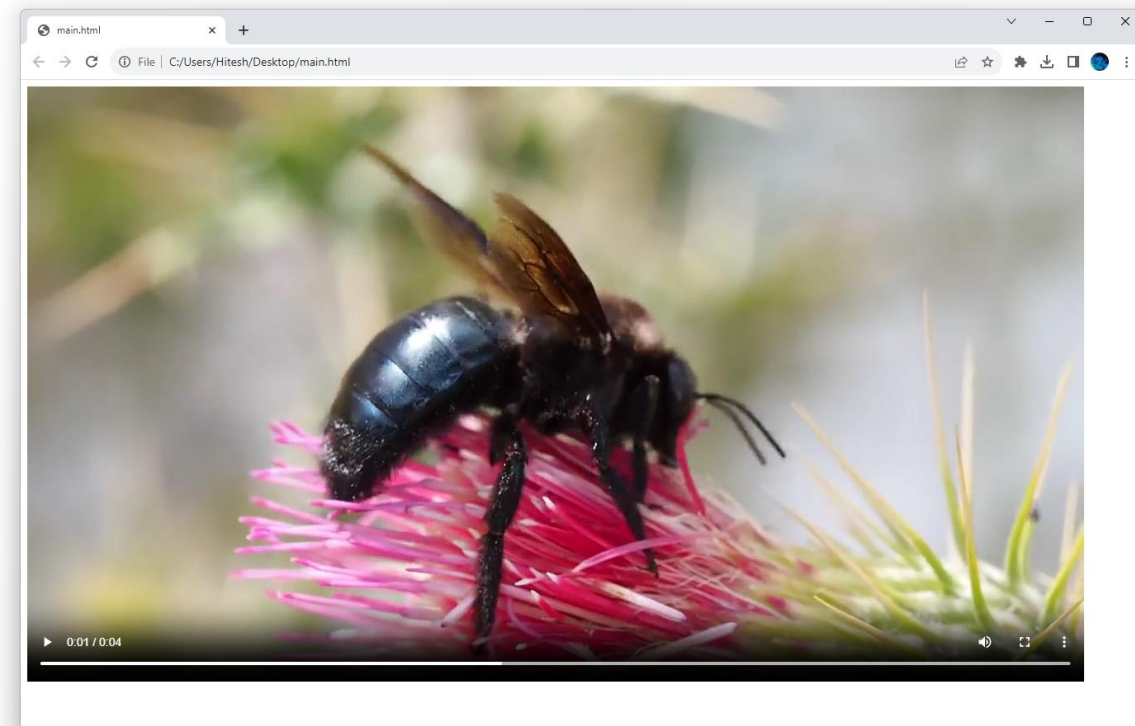
ANSWER: Compression
                  Hardware Acceleration
                  ...

Introduction:
Video playback in web-browsers.

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <video controls src="static/bee.mp4">
  </body>
</html>
```

- All modern web-browsers e.g. Google Chrome, Mozilla Firefox & Microsoft Edge etc. use FFmpeg under the hood for all kind of video/audio!

- https://www.chromium.org/audio-video/

- Key Considerations:
  - Magnitude
  - Performance
  - Stability
  - Range
  - ...

Table of Contents

- Multimedia
  - Types: Video, audio & image.
  - ~~Terms: Codecs, formats & containers.~~
  - Compression: Lossy & lossless.
  - FFmpeg: Importance, involvement & open-source.
  - ~~Video: Pixel, frame & representations.~~
  - Video: Typical video playback process.
  - Video: Hardware Acceleration, Software Vs. Hardware Decoding/Rendering
  - ~~Projects: Web-browsers (Google Chrome, Mozilla Firefox), VLC, mpv etc.~~
  - ~~Services: YouTube, Amazon Prime, Disney+Hotstar etc.~~
- Graphics
  - Differences/Similarities from "normal programming".
  - Graphics APIs: OpenGL, Direct3D, Vulkan, METAL.
  - "Hello Triangle!" in OpenGL.
  - Game engines: Unity, Unreal etc.
- Hardware
  - Why we need GPU?
  - CPU Vs. GPU
  - Speed differences & optimizations due to specific use-case.

Multimedia: Types

- Video
  - Moving visual content; generally combined with audio.
  - Formats:
    - HEVC (H.265), H.264, AV1, VP9, VP8 etc.
    - *Where is MP4 or MKV?* **Formats & Containers** *are different!*

- Image
  - Static visual content.
  - Formats:
   - JPEG, WEBP, PNG, BMP, TIFF, GIFF etc.

- Audio
  - Audible content.
  - Formats:
   - MP3, AAC, OPUS, Vorbis, FLAC, ALAC etc.

Compression: Lossy & lossless

- Lossy compression
  - The resulting file size is reduced.
  - Some information from the original file is lost e.g. quality of video or audio.
  - *e.g. MP3, JPEG etc.*
- Lossless compression
  - The resulting file size is reduced.
  - No information is lost.
  - e.g. FLAC, BMP etc.

Compression: Intra-frame & Inter-frame (video-specific)
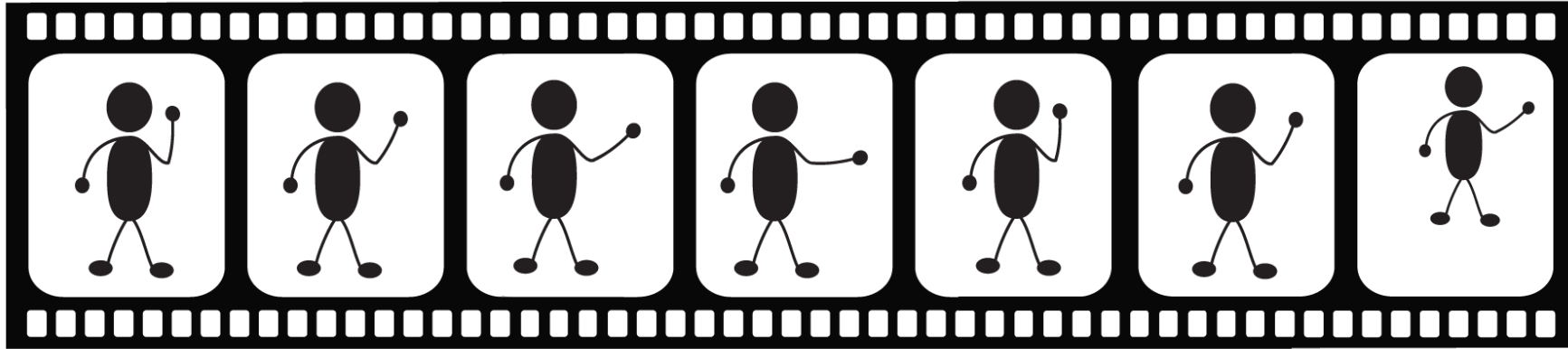
Intraframe Compression
Every frame is encoded Individually
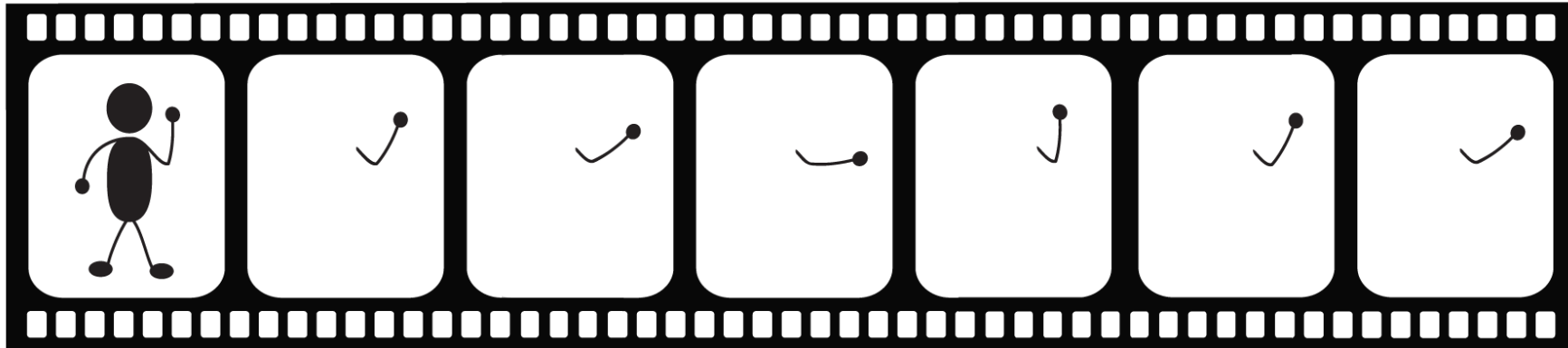
Interframe Compression
Only the differences between frames are encoded for each group of frames

FFmpeg

- Allows to stream, play, encode, decode, convert (or just about anything else) a piece of multimedia content i.e. video, audio or image.

- Supports all multimedia codecs, formats & protocols on the planet.

- Powers large amount of software & used by big-tech.

- Written in C & ASM to avoid OS-level or platform-specific abstractions i.e. cross-platform.

- Free & open-source. Developed under the GNU Lesser General Public License (LGPL).

- May be used as a standalone CLI application or C library.

- www.ffmpeg.org

- Tools: `ffmpeg, ffprobe, ffplay`

FFmpeg

FFmpeg

- `libavcodec`: provides implementation of a wider range of codecs.
- `libavformat`: implements streaming protocols, container formats and basic I/O access.
- `libavutil`: includes hashers, decompressors and miscellaneous utility functions.
- `libavfilter`: provides means to alter decoded audio and video through a directed graph of connected filters.
- `libavdevice`: provides an abstraction to access capture and playback devices.
- `libswresample`: implements audio mixing and resampling routines.
- `libswscale`: implements color conversion and scaling routines.

CLI

- `$ ffmpeg -i input.mp4 output.avi`
- `$ ffmpeg -i input_video.mp4 -vn -acodec mp3 output_audio.mp3`

C API

```
#include "libavformat/avformat.h"
#include "libavcodec/avcodec.h"
#include "libavfilter/avfilter.h"
#include "libavutil/avutil.h"
#include "libswresample/swresample.h"
```

Video: Typical video playback process.

- Protocol
  - A protocol refers to the method or rules used for communication between different components, such as fetching multimedia data from a specific source like a file, network stream, or device.
  - Examples: HLS, RTSP, RTMP, HTTP, HTTPS, PIPE, FD, FILE.
- Parser
  - A parser is responsible for interpreting the structure of multimedia data, such as breaking down a multimedia file into its constituent parts, like video frames and audio samples. It helps in understanding the format and extracting relevant information.
- Demuxer
  - A multimedia file generally contains one or more streams e.g. a video file may contain multiple video, audio or subtitle tracks.
  - The demuxer separates the combined audio and video streams from a multimedia container (e.g., MP4, MKV) into individual streams. It extracts the different elements (audio, video, subtitles) for further processing.
- Decoder
  - Decoding involves converting compressed audio or video data into a format that can be played or manipulated. Decoders are responsible for decompressing the multimedia data so that it can be processed for playback or editing.
- Renderer
  - The final step to display the decoded video frame (frame-buffer or texture) or audio frame (PCM).
  - The renderer is the component responsible for displaying or playing back the decoded multimedia content. It takes the processed audio and video data and outputs it to the display or audio playback device.

- **NOTE:** During FFmpeg compilation, one can manually select the supported protocols/parsers/demuxers/decoders.

Video: Hardware Acceleration, Software Vs. Hardware Decoding/Rendering

- "Hardware acceleration" from Google:
  - Hardware acceleration is the use of computer hardware designed to perform specific functions more efficiently when compared to software running on a general-purpose central processing unit.

- What does it mean in context of video playback?
  - CPU is good at general-purpose computation e.g. if, else if, for, while.
  - CPU is not good at handling/transforming/processing large amount of data as a video frame buffer.
  - GPU is excellent & far more efficient (speed & energy consumption) at this job.

  - We can still make CPU do GPU's job (recall it's a general-purpose computation). Thus, software-decoding/rendering. However, that results in decreased performance & very-high power consumption (less battery-life). It is near impossible to go beyond 1080p.

  - Most SoC/chipset have built-in hardware decoders for most codecs e.g. H.264, H.265, AV1.

Video: Hardware Acceleration, Software Vs. Hardware Decoding/Rendering

- Specification of MediaTek Dimensity 9000:
  - https://www.mediatek.com/products/smartphones-2/mediatek-dimensity-9000

Processor

Processor
- 1x Arm Cortex-X2 at 3.05GHz
- 3x Arm Cortex-A710 up to 2.85GHz
- 4x Arm Cortex-A510 up to 1.8GHz

Cores
Octa (8)

CPU Bit
64-bit

Heterogeneous Multi-Processing
Yes

Memory and Storage

Memory Type
LPDDR5x

Max Memory Frequency
7500Mbps

Storage Type
UFS 3.1

Connectivity

Cellular Technologies
2G-5G Multi-Mode, 5G/4G CA, 5G/4G FDD / TDD, CDMA2000 1x/EVDO Rev. A (SRLTE), EDGE, GSM, TD-SCDMA, WDCDMA

Camera

Max Camera Sensor Supported
320MP

Max Video Capture Resolution
4K (3840 x 2160)

Camera Features
9Gbps 18-bit HDR-ISPs / 4K Video HDR + AI-NR / Video Bokeh / Video EIS / AI-Shutter / AI-AE / AI-AF / AI-AWB / AI-NR HDR / AI-HDR / AI-FD

Graphics

GPU Type
Arm Mali-G710 MC10

Video Encoding
H.264, HEVC

Video Playback
H.264, HEVC, VP-9, AV1

Display

Max Refresh Rate
FHD+ @ 180Hz / WQHD+ @ 144Hz

AI

MediaTek 5th generation APU
APU 590 (multi-core)

Graphics: Introduction

- To make use of GPU's capabilities, conventional programming language (alone) isn't enough.
- Different platforms provide different abstractions in form of APIs to access GPU:

Microsoft Windows	Direct3D / DirectX
Apple macOS / iOS	METAL
Android	OpenGL ES
Cross-Platform	OpenGL, Vulkan

- OpenGL is most "widely accepted" and also works on Windows & macOS. However:
  - OpenGL on Windows is not performant (issue with manufacturer drivers & first-party support from Microsoft).
  - Apple deprecated OpenGL since macOS 10.14 Mojave.

Graphics: "Hello Triangle!" in OpenGL

https://learnopengl.com/Getting-started/Hello-Triangle

Every piece of content is essentially made up of triangles.

A number of concepts are involved:

- VBO: Vertex Buffer Object
- FBO: Frame Buffer Object
- Shaders:
    - Vertex Shader
    - Fragment Shader
- API provided by OpenGL:
    - glGenVertexArrays
    - glGenBuffers
    - glVertexAttribPointer
    - glBindVertexArray
    - glClear
    - glDrawArrays
    - glShaderSource
    - glCompileShader
    - glAttachShader
    - glLinkProgram

- Most game-engines e.g. Unity, Unreal, COCOS2D etc. provide abstraction.
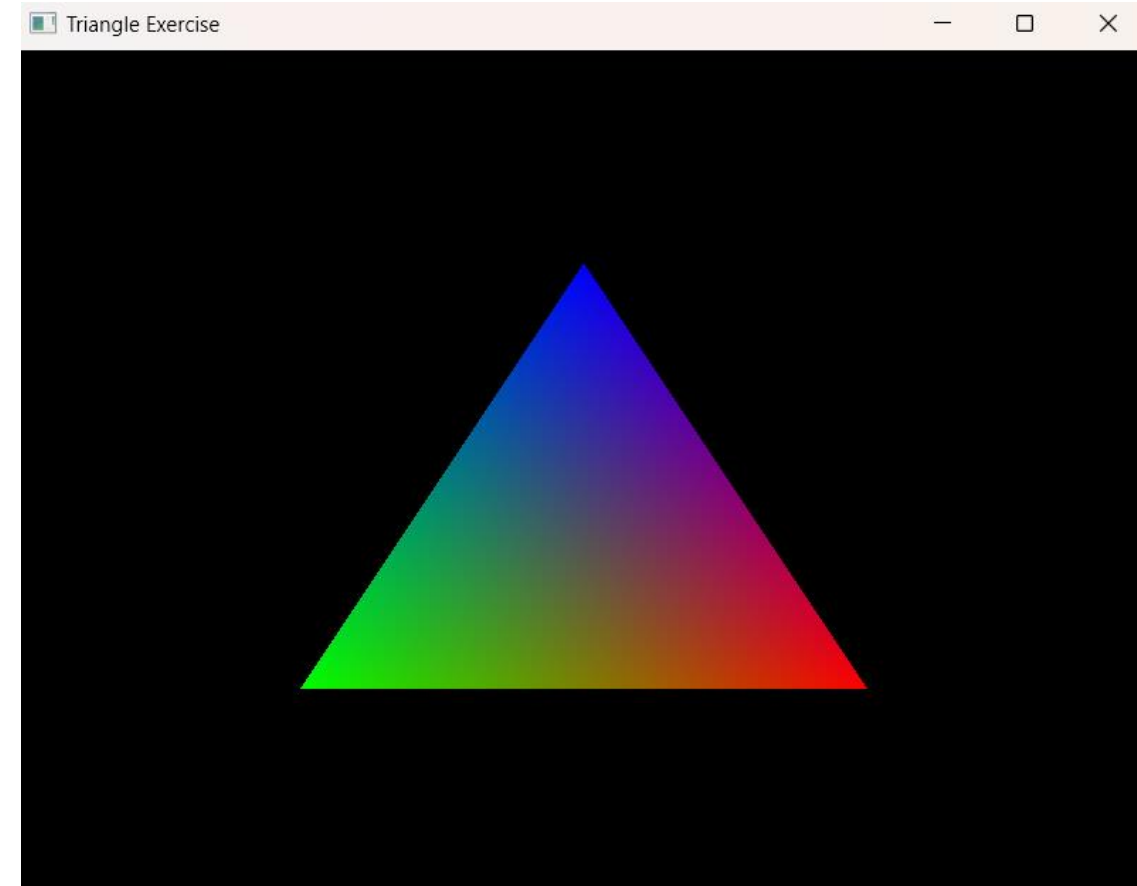
Graphics: ANGLE

- ANGLE: Almost Native Graphics Layer Engine
- Free & open-source. Developed under the BSD 3-Clause License.
- Started-off by Google. Contributed by Microsoft, Apple, Intel, ARM, Samsung, Adobe etc.
- ANGLE internally translates OpenGL ES to underlying platform's hardware-supported API i.e. DirectX on Windows, METAL on macOS etc.
- Simpler terms:
  Use OpenGL ES on every-platform without (*very-little*) compromising performance.

- https://github.com/google/angle

Graphics: ANGLE

Software utilizing ANGLE [edit]

ANGLE is currently used in a number of programs and software.

- Chromium and Google Chrome.[9] Chrome uses ANGLE not only for WebGL, but also for its implementation of the 2D HTML5 canvas and for the graphics layer of the Google Native Client (which is OpenGL ES 2.0 compatible).[8]
- Safari web browser uses ANGLE as basis for its WebGL implementation.[12]
- Firefox uses ANGLE as the default WebGL backend on Windows.[9]
- Qt 5 uses ANGLE as the default renderer for its OpenGL ES 2.0 API wrapper and other Qt elements which use it on Windows.[10]
- Godot uses ANGLE as an option for compatibility renderer for Windows and MacOS platforms starting with Godot 4.2[16][17]
- *Candy Crush Saga* uses ANGLE as the default renderer in its Windows Store version of the application.[11]
- Cocos2d uses ANGLE as its rendering engine for applications published to the Windows Store.[11]
- ANGLE for Windows Store[18] provides Windows developers precompiled ANGLE binaries via a NuGet package.
- Stellarium provides two versions for Windows: the default version uses OpenGL, the alternative version uses ANGLE as the renderer.[19]
- *Shovel Knight* uses ANGLE as rendering engine, as seen in final credits.[20]
- *RuneScape* NXT client uses ANGLE to provide a DirectX 9 compatibility mode for older graphics cards.[21]
- Krita started using ANGLE as the rendering engine on Windows starting on version 3.3.0.[22]
- Microsoft Edge has ANGLE as a rendering option in the "Standards Preview" page in Windows Insider build 17025.
- *Grand Theft Auto V* included ANGLE in the installation, normally at Systemdrive.
- OpenRA uses ANGLE for rendering on Windows[23]
- SolveSpace uses ANGLE on Windows.
- GameMaker: Studio uses ANGLE at compile-time to convert GLSL ES shaders to HLSL9 for the old Windows 32-bit export module.

Personal Project: media_kit

- https://github.com/media-kit/media-kit

Personal Project: media_kit
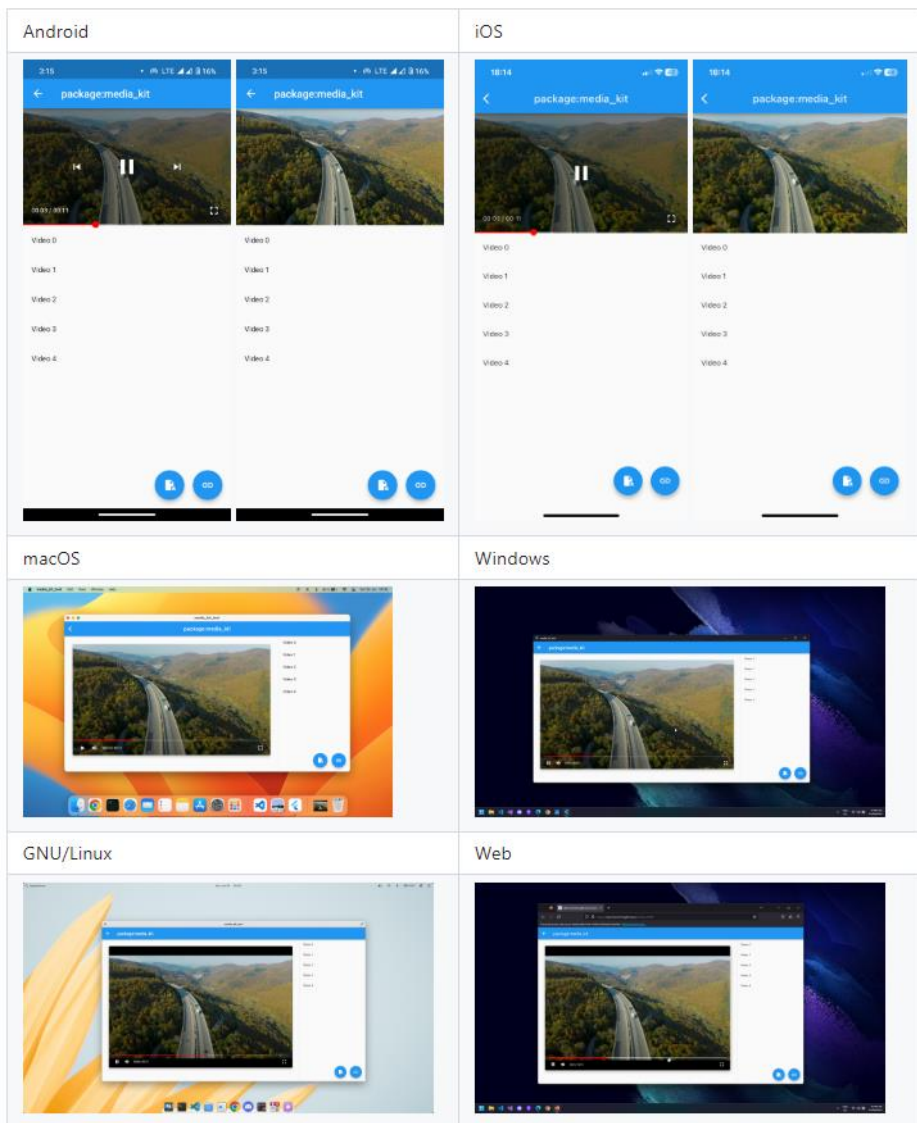
Following are known applications & projects that are using package:media_kit:

Aves	Aves is a gallery and metadata explorer app, built for Android with Flutter.
ente	ente is a cloud based mobile and desktop photo storage app with a focus on security and privacy.
Flarte	A desktop application to browse arte.tv.
Harmonoid	Plays & manages your music library. Looks beautiful & juicy.
MusicPod	Music, podcast and internet radio player for your Ubuntu Desktop.
PiliPila	A third-party application for BiliBili.
S3Drive	Zero Knowledge E2E encrypted storage compatible with any S3 provider.
Spotube	A fast, modern, lightweight & efficient Spotify Music Client for every platform.
Visual Physics	Physics made super-easy for JEE Main & Advanced, NEET.

Thank You

Questions?

Hitesh Kumar Saini

56195

 @alexmercerind
 @alexmercerind