Vulkan, OpenGL and/or Zerocopy

Matthew Waters (ystreet00)
GStreamer conference 2016
10 October 2016

Centricular

Who Am I

- Australian
- Work - Centricular
- Graphics – OpenGL, Vulkan
- Multimedia

Centricular

Quick Introduction – OpenGL

- OpenGL born from SGI in 1990's
- Cross-platform 3D API
  - X11, wayland, OS X, iOS, Android, Windows, Embedded Linux

Quick Introduction – GStreamer OpenGL/ES

- Minimum target OpenGL ES 2.0
  - Essentially the beginning of GLSL support
- Versions supported – OpenGL ES 2.0/3.x Desktop 2.x/3.x/4.x
- Platforms supported – Linux (X11 + Wayland), OS X, Windows, iOS, Android, Embedded Linux
- Various elements available – glimagesink, glcolorconvert, glvideomixer, gltransformation, gloverlay, gleffects_*, etc

Centricular

What's New? - libgstgl

- glviewconvert now supported on OpenGL ES 2.0 platforms (stereo elements as well)
- glcolorconvert now allows converting to multi-planar colorspaces in OpenGL ES 3.x
- gldownload API removed from the library
- dma-buf GL uploader
- GL queries
- Delayed GStreamer debugging

What's New? – libgstgl GL memory

- New GL buffer based GstMemory
- GstGLMemoryPBO – GL textures with Pixel Buffer Objects
- GstGLMemoryEGL – GL textures with EGLImage’s
- New GL renderbuffer based GstMemory
- GstGLFramebuffer

Centricular

Quick Introduction – Vulkan

- Vulkan released February 2016
- Cross-platform 3D API
  - X11, wayland, Android, Windows, Embedded Linux
- Aims to be a better fit for modern GPUs
- More control over synchronization

Vulkan in GStreamer

- vulkansink and vulkanupload elements
- Only basics implemented
  - Modelled on libgstgl API
- Much more work needed to be on par with OpenGL support

Vulkan in GStreamer

- Somewhat similar infrastructure to libgstgl
  - GstContext
  - Display
  - Window
  - Instance
  - Device
- Some things are very different
  - Synchronization – semaphores, events, fences
  - More application state

Centricular

Zerocopy - Introduction

- Definition ranges from:
  - No copies at all
  - No CPU performed copies
- Most common between decoder and renderer (but also occur between capturer/encoder)

Zerocopy – Where?

- VA-API
- OpenMAX IL
- Android’s MediaCodec
- iOS/macOS VideoToolbox
- VDPAU
- XvMC/XvBA
- DXVA

Zerocopy – How?

- Bind
  - VM MMU
  - Synchronization
- Use
- Unbind
  - Synchronization

Zerocopy – dma-buf

- Used by v4l2, (VA-API on EGL)
- dma-buf is converted to an EGLImage
- EGLImage is bound to a GL texture with glEGLImageTargetTexture2D()

Zerocopy – Android MediaCodec

- Hold the consumer end of a queue
- Can only pop off the end of the queue but don’t have unique frame handles
- Synchronization is all internal
- Can only bind to one GL context at a time

Zerocopy – OpenMAX IL

- Uses EGLImage’s
- RPi has a separate omx component – egl_render
- EGLImage backed by a GL texture passed into OMX
- Custom GstGLMemory

Centricular

Zerocopy – VideoToolbox - macOS

- Uses IOSurface
- CGLTexImageIOSurface2D()
- Custom GstGLMemory subclass

Zerocopy – VideoToolbox - iOS

- IOSurface is available but not public
- Uses CVOpenGLESTextureCache instead
  - CVOpenGLESTextureCacheCreateTextureFromImage()
- Custom GstGLMemory subclass

Zerocopy – gstreamer-vaapi

- GstVideoGLTextureUploadMeta
  - X11 uses GL_EXT_texture_from_pixmap
  - Wayland/EGL uses dma-buf

What's Happening?

- OpenGL helper library – move to gst-plugins-base
- OpenGL model viewer – still :-)

Centricular

Thanks!

ystreet00 in #gstreamer on freenode

Centricular