

# Multimedia, PDF & ConTeXt

Michal Vlasák, 2021

# Overview

- Inserting multimedia into PDFs with ConT<sub>E</sub>Xt
- PDF multimedia / interaction concepts
- Three aspects:
  - PDF standard
  - PDF viewers
  - ConT<sub>E</sub>Xt
- Various PDF mechanisms
- Future development

# Multimedia

- audio (mp3, ...)
- video (mp4, ...)
- 3D files (u3d, ...)

# Evaluating the mechanisms

- Criteria:
  - support in PDF standards
  - supported media types
  - supported source types
  - goodies (parameters, controls, scripting)
  - support in PDF viewers
  - support in ConT<sub>E</sub>Xt

# Files

- Three basic types:
  - Embedded file
    - file is *in* the PDF as stream
  - External file
    - relative / absolute path
  - URL file

# Annotations

- (dumb) rectangular area on some page
- meaning depends on *type* of the annotation
  - Screen annotation – area of media playback
  - Link annotation – active area
  - Square, Circle annotations – icons
- “markup” vs “primary”

spanning two lines.

A long text

# Actions

- ConT<sub>E</sub>Xt references
- different types:
  - “Jump actions” (go to different part of this document, go to URL, ...)
    - `\goto{text}[reference]`
  - Multimedia actions
    - `StartRendering`, `StopRendering`
  - Special actions
    - `NextPage`
- different means of execution
  - initial (“open”) action
    - `openaction` in `\setupinteraction`
  - Link annotation
    - `\goto`
  - Trigger events
    - page open / close

# Sound

- since PDF 1.2 (1996) until PDF 1.5 (2003) / PDF 2.0 (2017)
- encoded audio file / raw audio
- external file (for files), embedded file (raw audio)
- parameters (volume, repeat, mix), appearance: speaker / microphone icon, sound actions
- raw audio in Acrobat
- not in ConT<sub>E</sub>Xt



# Movie

- since PDF 1.2 (1996) until PDF 1.5 (2003) / PDF 2.0 (2017)
- encoded video file
- any source (embedded, external, URL file)
- parameters (start, stop, rate, volume, controls, repeat), “poster” (custom or implicit), actions
- Evince and Okular

```
\externalfigure[video.mp4]  
  [width=\textwidth,  
  height=.461\textwidth]
```

```
\useexternalsoundtrack  
  [myaudio] [audio.mp3]  
\checksoundtrack{myaudio}
```

# Multimedia (“Renditions”)

- since PDF 1.5 (2003), “legacy” in Acrobat
- any format (e.g. Flash, image), audio video in practice
- any source (embedded, external, URL file)
- plugin based, media selection, screen/rendition split, actions *and* JS scripting, parameters (start, stop)
- Acrobat, Foxit, Okular, Evince

```
\useexternalrendering[embedded]  
  [video/mp4] [video.mp4] [embed=yes]
```

```
\useexternalrendering[URL]  
  [video/mp4]  
  [https://example.com/video.mp4]
```

```
\useexternalrendering[external]  
  [video/mp4] [video.mp4]
```

```
\definerenderingwindow[mywin]  
  [width=\textwidth,  
  height=\textwidth]
```

```
\placereadingwindow[mywin] [embedded]  
\placereadingwindow[mywin] [URL]  
\placereadingwindow[mywin] [external]
```

# 3D art

- since PDF 1.6 (2004)
- U3D, PRC
- stream source only
- very interactive, highly customizable (camera position/direction, background color, lighting, ...), JS scripting and PDF actions
- Acrobat, Foxit (no JS nor actions)

```
\useexternalfigure  
  [cloudq]  
  [cloudq.u3d]  
  [width=0.7\textwidth,  
  height=.7\textwidth,  
  display=u3d:myset:display:1,  
  controls=u3d:myset:controls:1]
```

# Rich Media

- since “PDF 1.7+” (2008) or PDF2.0
- audio, video, 3D, Flash
- embedded files only
- windows, Flash media players, more scripts (3D), no controls, parameters
- Acrobat, Okular (Flash only)

# The distribution of knowledge

- PDF stores either:
  - raw data
  - file format
- metadata location
- compression
  - PDF compression vs file format)
- complexity
  - in PDF viewer *and* writer or in libraries

```
<<
  /Type /XObject
  /Subtype /Image
  /Width 640
  /Height 400
  /BitsPerComponent 8
  /ColorSpace /DeviceRGB
  /Length 62163
  /Filter /FlateDecode
  /DecodeParms <<
    /Colors 3
    /Columns 640
    /BitsPerComponent 8
    /Predictor 10
  >>
>>
stream
[...]
endstream
```

# GStreamer

- <https://gstreamer.freedesktop.org/>
- “library for constructing graphs of media-handling components”
- sources (file, http, alsa, ...), formats (mp4, opus, ...), filters, outputs (alsa, file, ...)
- based on plugins – good, bad, ugly, others
- Recommended packages for Linux:
  - gstreamer
  - gst-plugins-base
  - gst-plugins-good
  - gst-plugins-ugly
  - gst-libav

# References

- `interaction.pdf`
- Renditions
  - `scrn-wid.mk1x (examples)`, `scrn-wid.lua`, `mk1x/lpdf-wid.lmt`
- 3D art
  - `back-u3d.mk1x (examples)`, `grph-u3d.lua`, `lpdf-u3d.lmt`
- Movies
  - `lpdf-mov.lmt`, `grph-inc.lmt`
- Rich Media
  - `back-swf.mkiv (examples)`, `lpdf-swf.lua`, `grph-swf.lua`

# What now

- Current state
  - several flawed / unsupported mechanisms out there
  - Flash is really dead
- Ways forward
  - give up completely
  - improve ConT<sub>E</sub>Xt
    - new backing mechanism for `\externalfigure`?
    - support more options for Renditions?
    - support plain Rich Media?
  - improve PDF viewers
    - Okular, Evince (GStreamer)
    - a new mechanism?
      - Movies + JavaScript API



# Spare time

- Demos, test files
- thesis
- pdfextra