

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are positioned diagonally, with the blue one partially covering the green one.

# Video Player Research

By Arnau Lucena Cobo.



# Index

- 1- What is a video player?
- 2- Types of cutscenes in video games.
- 3- Video Files & Codecs.
- 4- Implementation in code.
- 5- Possible improvements.
- 6- Exercises.
- 7- Sources.

# 1- What is a Video Player?

- A video player is a kind of media player for playing back digital video data from media.
- Many video players support simple playback of digital audio & 3D playback of 2D video.
- When it comes to videogames, these are used to:
  - Cinematically showcase situations that would not be shown efficiently through gameplay.
  - Avoid processing in real time.
  - Showcase animation prowess.
  - Help expand the author's creative options.



## 2- Types of cutscenes in video games

There are a total of five types of cutscenes used in video games:

- Live-action cutscenes.
- Pre-rendered cutscenes.
- Real time cutscenes.
- Mixed media cutscenes.
- Interactive cutscenes.



# Question



What kind of cutscene is this?

- Live-action cutscene.
- Pre-rendered cutscene.
- Real time cutscene.
- Mixed media cutscene.
- Interactive cutscene.

# Answer



What kind of cutscene is this?

- Live-action cutscene.
- ~~Pre-rendered cutscene.~~
- ~~Real time cutscene.~~
- ~~Mixed media cutscene.~~
- Interactive cutscene.

# Live-action cutscenes

- Live-action cutscenes have many similarities to films, using constructed sets & actors to portray the characters.
- Some games have extensively used film footage & other assets from the film production in their cutscenes.
- Detroit: Become Human's cutscenes are a great example of what they are.
- They are also interactive cutscenes. This will be explained later.



# Pre-rendered cutscenes



- Pre-rendered cutscenes are animated & rendered by the game's developers.
- They take advantage of the full array of techniques of CGI, cel animation or graphic novel-style panel art.
- The video I use in my project would count as a pre-rendered cutscene.





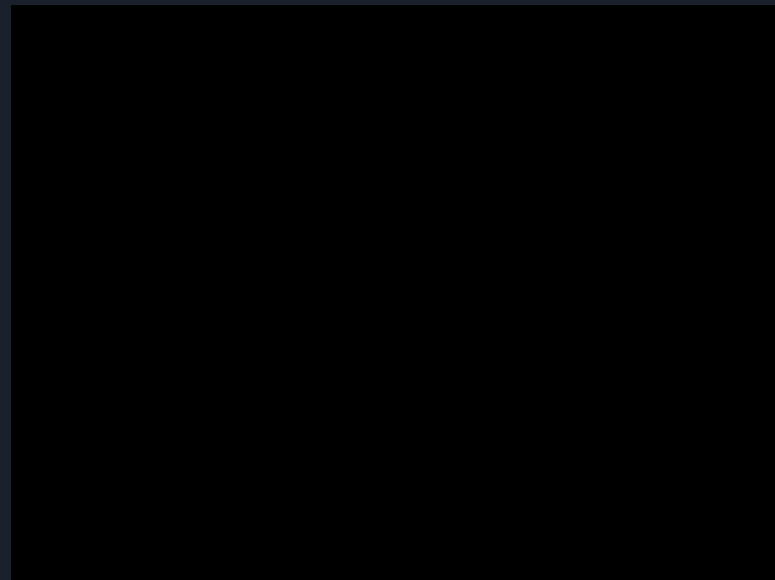


# Real time cutscenes

- Real time cutscenes are rendered on-the-fly using the same game engine as the graphics during gameplay.
- This technique is also known as Machinima/Machinema.
- These are usually of much lower quality than pre-rendered cutscenes, but can adapt to the game's state.

Example: Some games allow the player character to wear several different outfits, and appear in cutscenes wearing the outfit the player has chosen, and sometimes even give players control over camera movement during them.

# Real time cutscenes



# Mixed media cutscenes

- These are basically the combination of both pre-rendered and real time cutscenes on a game
- Many games use these, as the developers may feel it appropriate for scenes to be of one type or other.
- Genshin impact is a good example of mixed media cutscenes.



# Interactive cutscenes

- Interactive cutscenes, also known as Quick time events involve the computer taking control of the player character while prompts (such as a sequence of button presses) appear on screen, requiring the player to follow them in order to continue or succeed at the action.



- Detroit: Become Human fits in this category too.
- Other examples:
  - Games from Telltale Games.
  - Other stuff from David Cage.





# 3- Video files & Codecs

Video files are formats used to store digital video data on a computer system, usually consisting of a:

- Container (MP4, AVI, Webm...)
  - Video coding format (VP9, H.264, H.265...)
  - Audio coding format (MP3, AAC, Vorbis...)
  - Metadata & others.

Media players with high compatibility usually end up being a coding hell due to the fact that they have to keep in mind the many formats within the many containers.



# 3- Video files & Codecs

Format Container: .avi, .mp4, .mov, .ogg, .flv, .mkv, etc.

Video codec:

H.264,  
VC-1,  
Theora,  
Dirac 2.1,  
H.263,  
etc.

Audio codec:

AAC,  
WMA,  
Vorbis,  
PCM,  
etc.

Captioning,  
Video description:

SAMI, SMIL,  
Hi-Caption,  
CMML, DXFP,  
3GPP TS 26.245,  
MPSub,  
etc.

Metadata:

Author,  
Title,  
Location,  
Date,  
Copyright,  
License,  
etc.

# 3- Video files & Codecs





# 3- Video files & Codecs

- Each codec has its own license type.
- Therefore, developers need to be careful to avoid legal trouble.
- I shall now talk about the following codec options:
  - Bink.
  - Platform specific.
  - H.264.
  - VP8/VP9.



# 3- Video files & Codecs

- Bink:
  - Pretty much the de facto standard in the video game industry.
  - Used in more than 5800 games on 14 different platforms.
  - It has a pricing of \$8500 per game per platform.





## 3- Video files & Codecs

- Platform specific:
  - We use the video playing capabilities inherent in each platform (WIndows Media Foundation, QuickTime...).
  - Free to use.
  - Can give access to special video decoding hardware (H.264 hardware available in most phones)
  - Low CPU usage.
  - Not good for targeting multiple platforms.



# 3- Video files & Codecs

- H.264:
  - Most popular commercial codec.
  - Used in Blu-ray, video cameras, iTunes, Youtube...
  - Covered by patents. There is a patent pool containing over 1700 patents.
  - You could still get sued :)





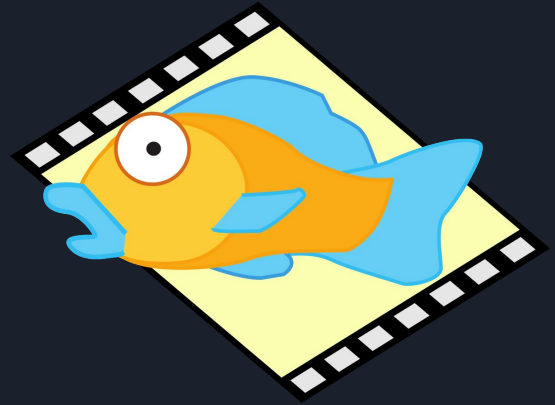
# 3- Video files & Codecs

- VP8/VP9:
  - Free video codecs owned by Google.
  - Free to use patents.
  - BSD licensed library libvpx for encoding/decoding video files.
  - There is constant discussion on whether it infringes or not on patents.
  - This usually happens with free codecs.

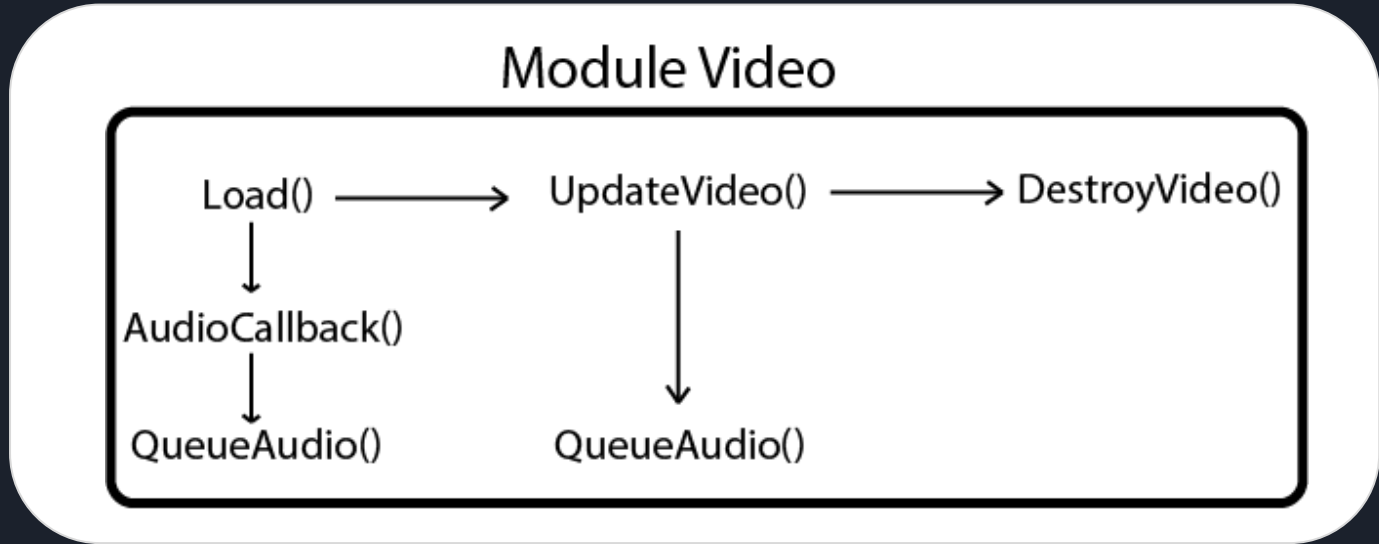
VP8  
VP9

# 4- Implementation in code

- The codec I have decided to use is Theora:
  - It's free!
- I will also be using Theoraplay:
  - Used to play encoded videos as `SDL_Textures`.
  - Free to use & unencumbered.



## 4- Implementation in code





# 5- Possible improvements

1. A pause function
2. A method of replaying the video or playing multiple videos (Even at the same time), as as of right now only one video can be played at a time and only once.
3. A method to scale videos regardless of quality, as the video player currently doesn't scale the video to the window.
4. A method to easily modify the volume, as it is currently set to a specific amount.



# 6- Exercises

TODO 1:

Use `THEORAPLAY_Decoder` & `THEORAPLAY_startDecodeFile` to decode the format.

Hint: `THEORAPLAY_startDecodeFile`(File path, number of frames(Set to 30), Format to decode (`THEORAPLAY_VIDFMT_IYUV`)).

TODO 2:

Create a video & audio buffer (`THEORAPLAY_VideoFrame` & `THEORAPLAY_AudioPacket`) and fill them with `THEORAPLAY_getAudio` and `THEORAPLAY_getVideo`.





# 6- Exercises

TODO 3:

In case of error, free the audio & video buffers & stop the decoding.

TODO 4:

Get & process the video data from the decoder.

TODO 5:

Just like the TODO 3, free the video & audio buffers & stop the decoding, but also destroy the texture.

TODO 6:

Use the video's load function and save it to the video variable (You will need the file pathing & the renderer)



# 7- Sources

<https://maciadalmu.github.io/Video-Player-Research/>

<https://axelalavedra.github.io/Video-Player-Research/>

<https://en.wikipedia.org/wiki/Cutscene>

<https://en.wikipedia.org/wiki/Codec>

[https://www.gamasutra.com/view/news/170671/Indepth\\_Playing\\_with\\_video.php](https://www.gamasutra.com/view/news/170671/Indepth_Playing_with_video.php)

[https://en.wikipedia.org/wiki/Video\\_file\\_format](https://en.wikipedia.org/wiki/Video_file_format)

<https://www.theora.org/downloads/>

<https://icculus.org/theoraplay/>

These were used as references for the code:

<https://glusoft.com/tutorials/sdl2/playing-theora-video>

<https://github.com/maciadalmu/Video-Player-Research>



# That is all!

Thank you for your attention!