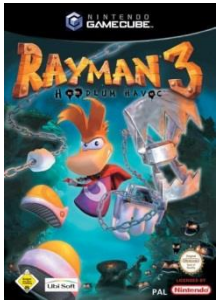
The Audio Callback for Audio Synchronization

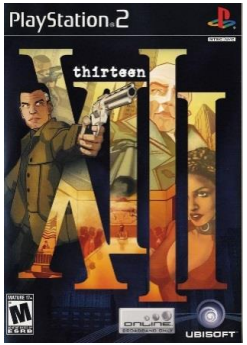
Mathieu Pavageau
Senior Sound Programmer
Ubisoft Paris Studio

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 25-29, 2013
EXPO DATES: MARCH 27-29
2013

Sound Programming Experience

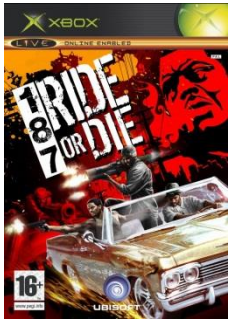


2003



2003

2004



2005

2011

...

+ Sound Middleware

+ R&D

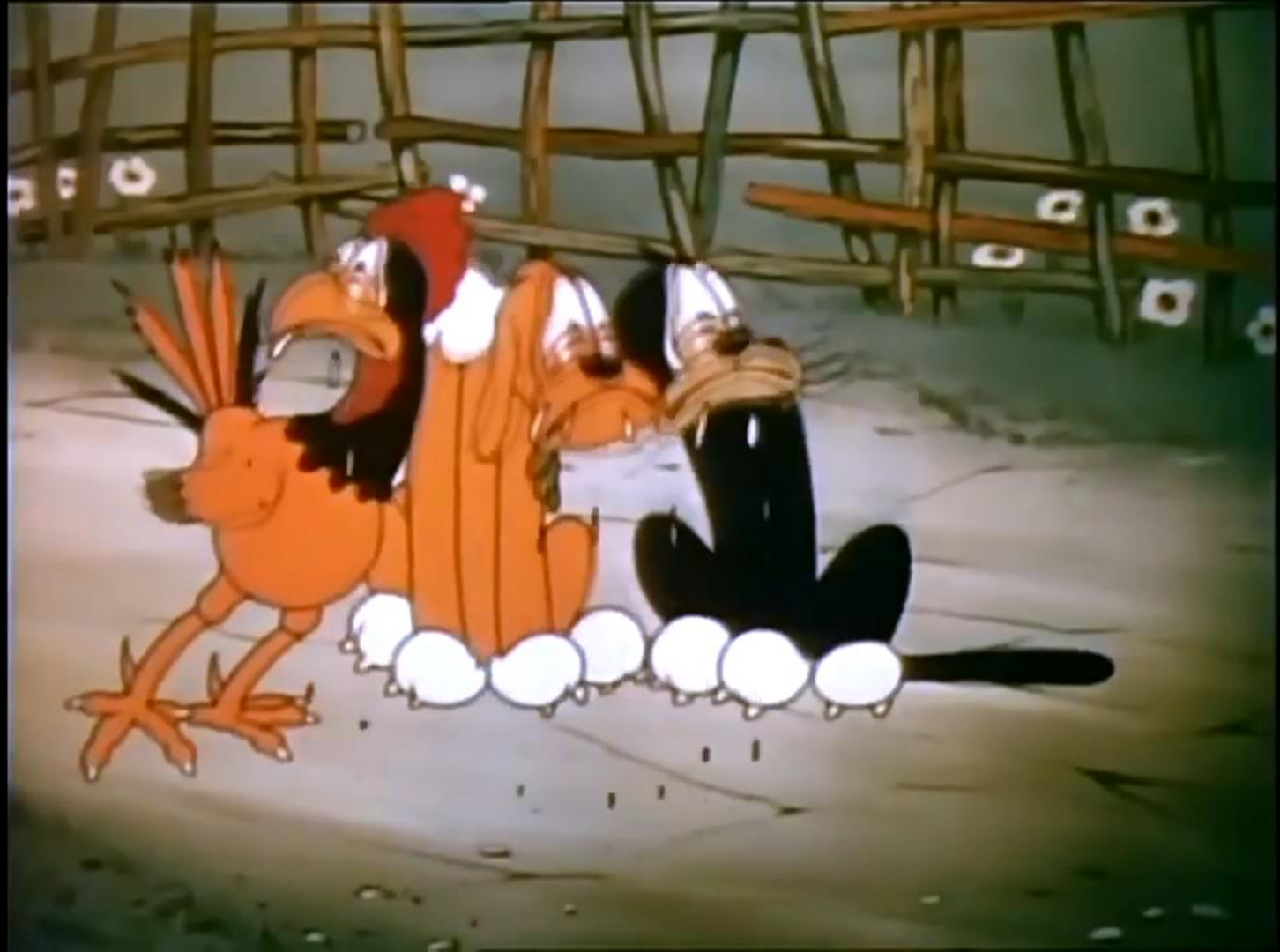
Outline

- Musical Interactivity
- Audio Engines in Games
- Audio needs a fast update context
- Implementing the Audio Callback
- Examples

Musical Interactivity

- Musical Content reacts to Gameplay
- Music and Gameplay designed together

Example:
Mickey-Mousing



Current Musical Interactivity

Either Pre-Recorded Audio

- One music track at a time
- Or multitracks/multilayers

High Quality Audio

BUT Little Musical Interactivity

Or MIDI System

- Musical Score can be modified in real-time
- Uses Sound Banks

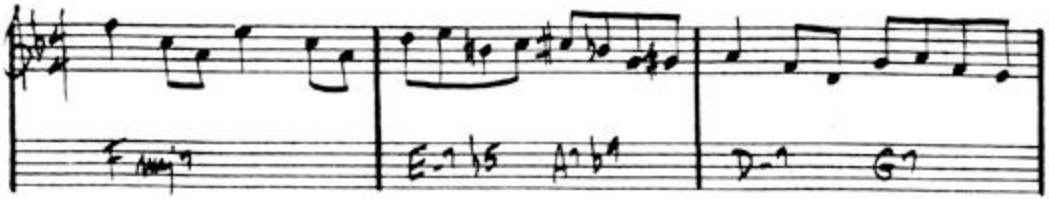
High Musical Interactivity

BUT Low Audio Quality

Synchronizing Audio with Audio

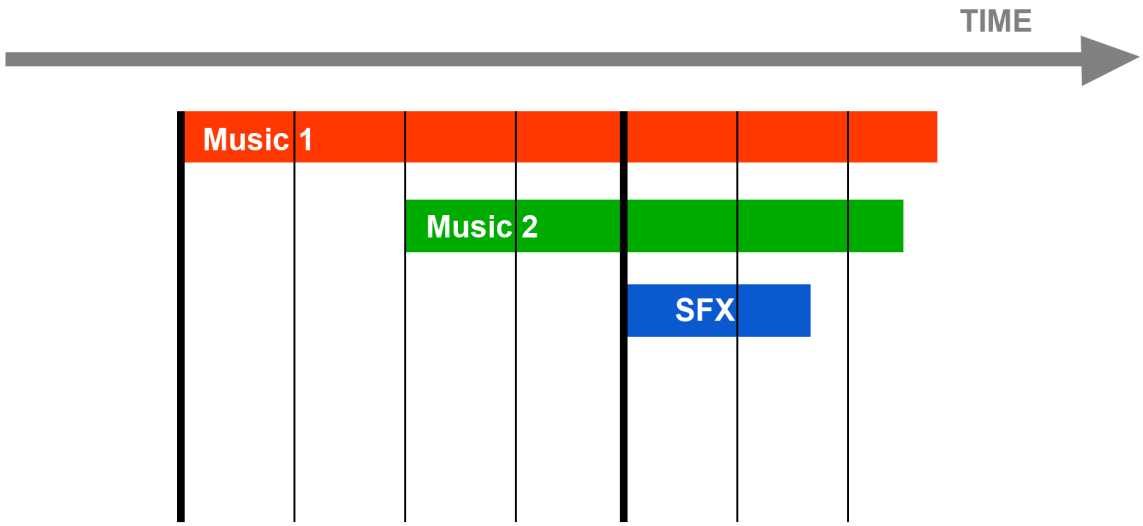
Synchronization: Design

- Musical Structure: data with the wave



- Play new sounds sync with playing sounds

Synchronization: Design



4:4 Music Structure

But: V-Sync is Master Clock

- V-Sync image update
- Video frame is Master Clock (FPS)
- Entire Game updated with V-Sync

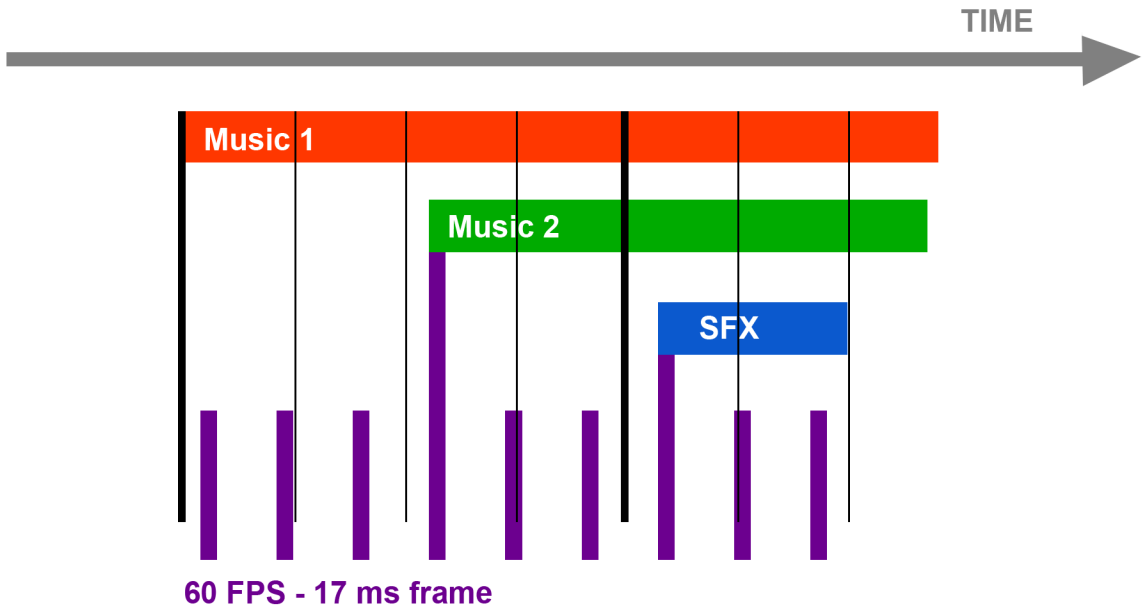
All Audio Requests made in Video Frame

Duration of Game Frame

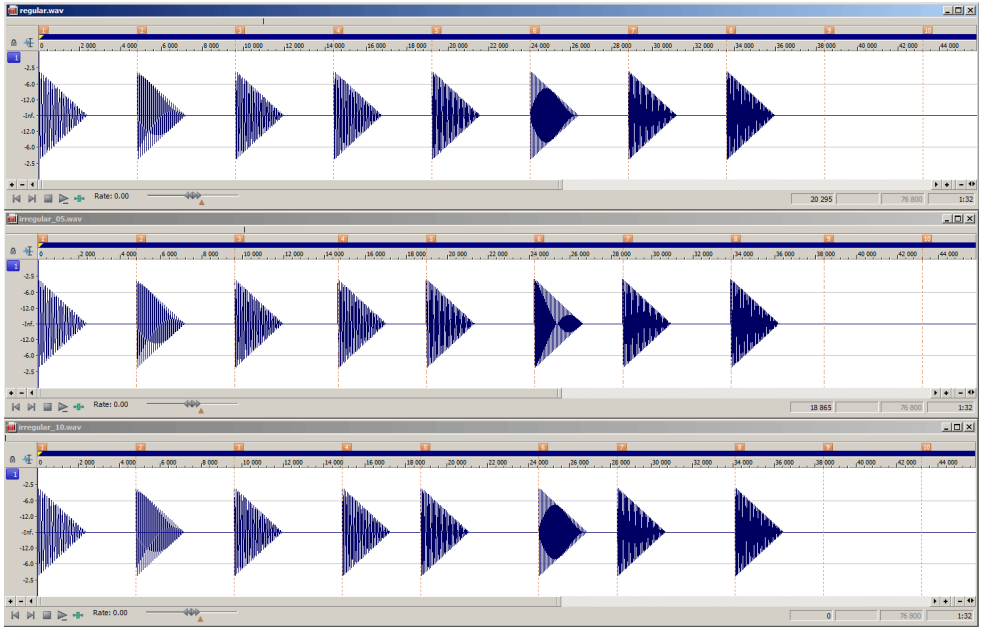
Frame Rate	Duration
60 FPS (60 Hz TV)	**17 ms**
50 FPS (50 Hz TV)	**20 ms**
30 FPS (60 Hz TV)	**33 ms**
25 FPS (50 Hz TV)	**40 ms**

17 ms is the best time precision for audio

Without the Audio Callback: No Sync



Time Perception Examples (1)



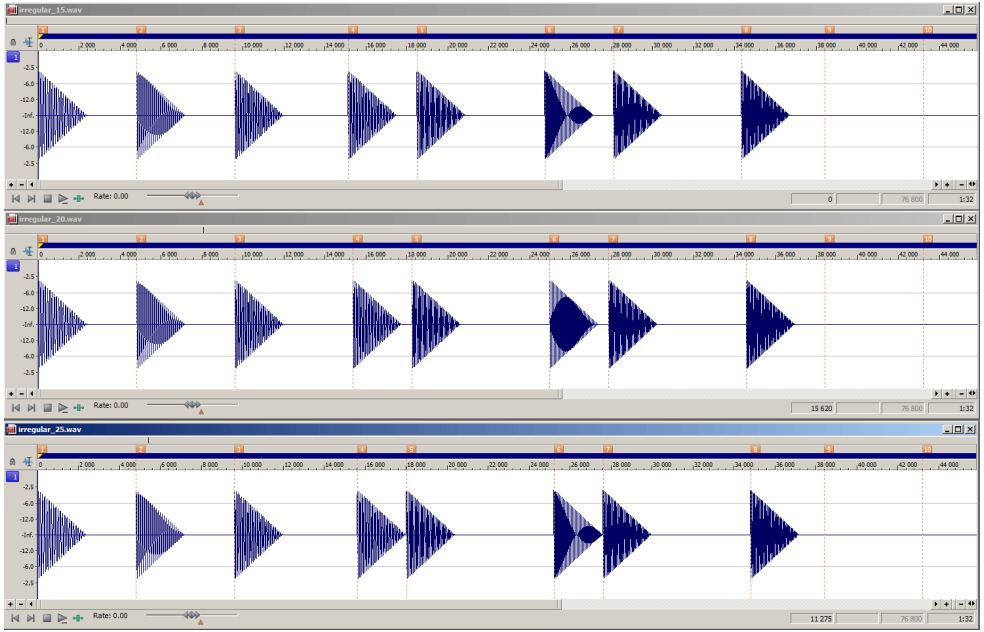
A B C D E F G A

0 ms

5 ms

10 ms

Time Perception Examples (2)



15 ms

20 ms

25 ms

A B C D E F G A

Time Perception for Audio

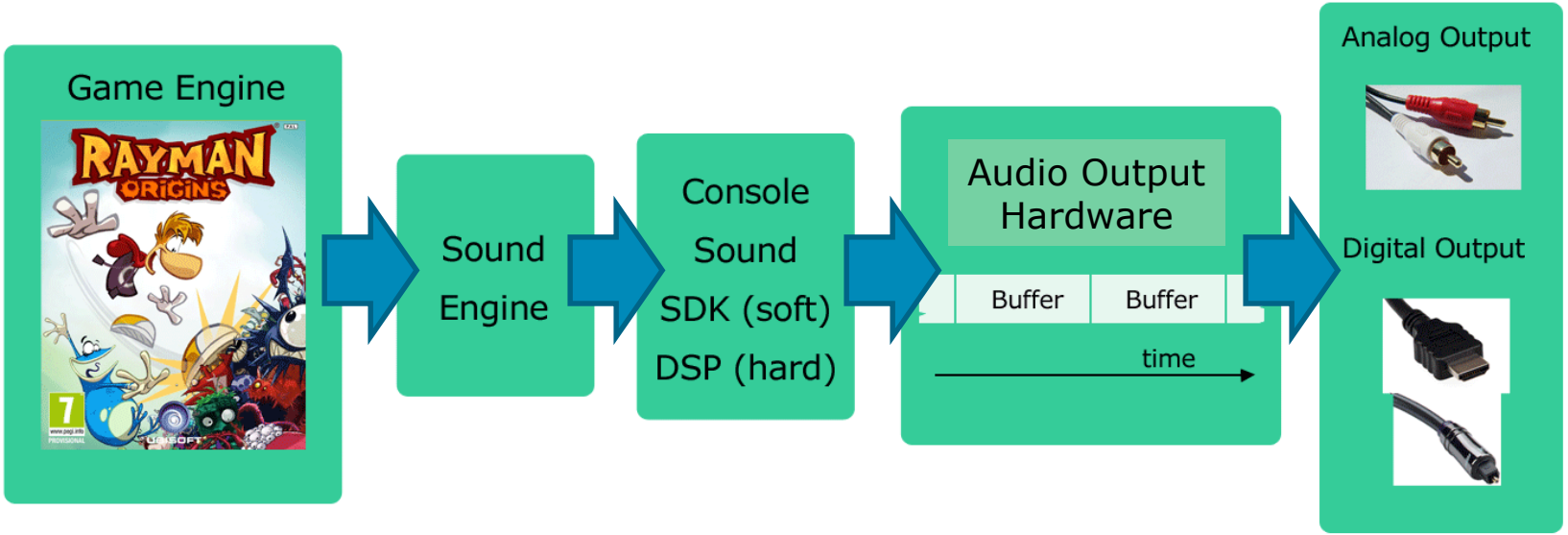
- Synchronized sounds require a precision of a few milliseconds

- Musical precision does not need sample to sample precision

Audio Synchronization
needs a fast update context:

The Audio Callback

The Audio Callback

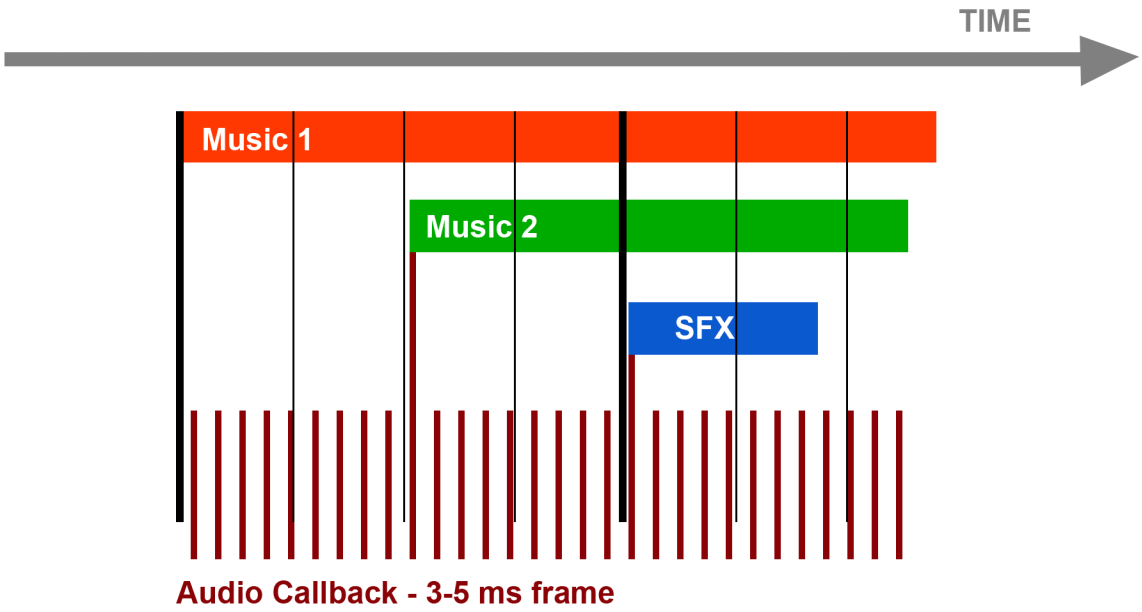


Regular filling-up of audio buffers for the hardware

Audio Callback on Consoles

- Wii/WiiU: AX callback (3 ms)
- PS3: audio update thread (5 ms)
- XB360: XAudio2 callbacks (5 ms)

With the Audio Callback: Audio Sync



Audio Callback Implementation

Implementation: Precautions

- Callback cannot be slowed down
- Lockless programming
- Some operations are forbidden (ex CreateVoice on XAudio2)

Architecture

- Low level audio must expose the Audio Callback to the game/client (for interactivity with the game)

- It’s difficult to modify existing sound engines to use the Audio Callback

Advantages

- Stable audio, not affected by FPS drop
- Behaves the same rhythmically when ported on different platforms

Examples in Ubisoft Games

Synchronized Play Requests

- Dynamic Multitracks

- PlayOnNextBar, PlayOnNextBeat, etc…

Example:
Rayman Origins Menu


Select a file.
0/246
0/10
New game
New game
Without sync
Y Delete file
Select A


Select a file.
0/246
0/10
New game
New game
Y Delete file
Select A

With sync

Example:
Rayman Origins Intro


37

Summary

- Audio synchronization can be implemented using the Audio Callback
- Audio synchronization enables better audio interactivity
- Audio interaction requires sharing code with the game: Architecture

What Next?

- Next Gen machines will give even more power to audio
- More process can be done in the Audio Callback (effects, software mix, entire DAW…)
- Mickey Mousing one day in video games!

Questions?

More info: mathieu.pavageau@ubisoft.com