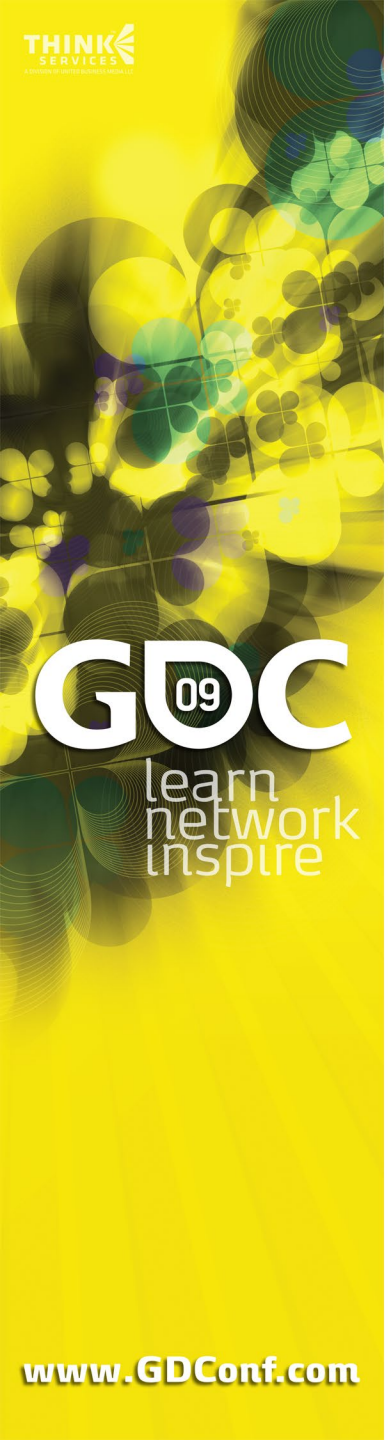

GDC
09
learn
network
inspire
www.GDConf.com
Game Developers Conference®
March 23-27, 2009 | Moscone Center, San Francisco



How High Dynamic Range Audio Makes Battlefield: Bad Company Go ”BOOM”

Anders Clerwall
EA/DICE

Overview

- Retrospective
- High Dynamic Range Audio
- Implications and Benefits
- Culling and Prioritizing
- Mixing & Mastering
- Conclusions
- Future

- Q&A

Retrospective

- Dynamic range traditionally faked
  - Duck and snapshot-mixing
  - Suits games with predictable sound scape
    - Still, a lot of manual labor
- ..Even for multiplayer
  - Highly unpredictabale sound scape
- Everything was turned to 11
  - ”Every sound is important!”
    - But not at the same time!
  - Amplitude 1.0 has no meaning

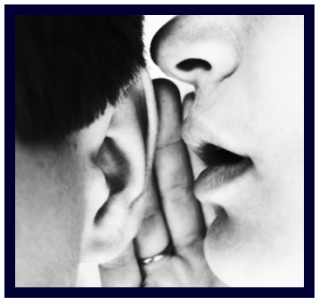
Retrospective

- Battlefield: Bad Company
  - DICE's first "Next-Gen" (360/PS3) game
  - More of everything
  - The sky's the limit!
- Frostbite
  - A new beginning
  - We decided to use HDR audio from the start!
    - Even the name *sounded* "Next-Gen"
  - It's all about the dynamic range!

What is HDR Audio?

- It is..
  - An automatic mixer
  - An automatic prioritization method
  - An effective culling method
  - A way of mapping one part of a large dynamic range to a smaller one (amplitude)
    - Basically like HDR lighting (Tone Mapping)
- It is not..
  - Compression
    - Although sometimes similar

What is HDR audio?



Great, but why?

- Sadly we can't play at 128dB in peoples' homes
  - Neighbors, hearing loss, broken windows
  - Technical issues with audio equipment
- Need a way to make sounds feel really loud
  - But still hear quiet sounds
  - Without all that traditional manual labor
- Spend our time more wisely

Great, but why?

- Lots of loud sounds make for a crowded upper part of the amplitude range
  - Difficult to make quiet sounds temporarily loud
- ”Every sound is important!”
  - But not at the same time!
  - So we need a way to tell which ones are

How did we do it?

- Specify loudness (dB_{SPL}) instead of amplitude
  - The key to HDR audio
  - Use real-life measurements as guideline
  - One value-range for all sounds
- Measure each sound at the listener position
  - Perceived loudness
  - 3D calculations
    - Angle and distance attenuation/filter coefficients, etc.
    - There’s no static ”MaxDistance” (gasp!)

How did we do it?

- Every sound contributes to the current loudness

$$L_{current} = 10 \times \log 10 \left(\sum_{n} 10 \times \frac{L_n}{10} \right) dB$$

- Position the HDR ”window”
  - This is the most important part of the dynamic range
  - Defined by MinTop, Size, ReleaseTime
  - Loudest sound can move the window
    - $L_{winTop} = \max\left(L_{current} \middle| L_{minTop}\right)$

How did we do it?

- Calculate amplitudes

$$A_{final} = \text{pow10}\left(\frac{L_{perceived} - L_{winTop}}{20}\right)$$

  - Apply headroom if needed
  - Amplitude 1.0 has actual meaning again!
- Let's listen to it!

Simple scenario video

Implications

- Less control
  - No guarantee that a particular sound can always be heard
  - No way of predicting how loud a sound will be, in absolute volume
- Somewhat of a paradigm-shift for sound designers
  - No longer controlling absolute volume, but rather the relative volume between sounds
  - Do they actually *need* control over that?
    - I'm not a sound designer ☺
    - They haven't complained (at least not about that)

Benefits

- *More* control
  - Loudness actually *means* something!
    - Preserves the relationship between sounds in a clear way
  - It's always obvious which sounds *should* play
- Based on reality
  - But actually *not* reality
  - We're making a game!
  - Creative control very important

Benefits

- Easier to make exceptions to the rule than adding lots of rules
  - For example; give first-person sounds and VO some headroom
- Less/cleaner code
  - Means less bugs. Promise!
- Mixing the game
- Enables some pretty nice culling schemes
  - Harsh, but fair

Pre-culling

- Before a sound gets activated and starts rendering
  - Start with the sound's specified loudness
    - Apply distance/angle attenuation
    - Remember, this is as loud as the sound could *theoretically* get!
  - Previous update's HDR window bottom is cutoff
- Helps clean up the sound scape
- Low overhead
  - These calculations would be done anyway

Pre-culling

- One-shots can be completely ignored
- Loops are considered infinite and become pending
  - Until they're explicitly stopped
- Shifts load-balancing to where most knowledge is
  - More reliable
  - Distance is inferred by loudness
    - It's about what you *hear*, not how *far away* you are

Culling of active sounds

- Every sound is either ”Audible” or ”Inaudible”
  - Basically above or below some threshold of hearing
  - Assume HDR window bottom
    - May end up being the *n*th sound’s perceived loudness due to load-balancing
- Once a sound becomes inaudible, suspend it
  - One-shots may never return
  - Looping sounds most likely will
  - Resume once audible again

Culling of active sounds

- Worst-case: Lots of similarly loud sounds
  - They’re all inside the current HDR window
  - Limit on number of active sounds affects threshold of hearing
- Best-case: Your weapon vs. bullet impacts nearby
  - Weapon will raise HDR window and bullet impacts can be completely ignored
  - Very common scenario in Bad Company

Better measurements

- Measure each sound's RMS
  - Apply to perceived loudness

$$L_{perceived} = L_{attenuated} + 20 \times \log10(A_{rms})$$

- Compensate for maximum sine RMS when pre-culling
  - Subtract 3dB from "attenuated" loudness
  - Due to peak-normalization in pipeline
- Improves culling and prioritization
  - Need to guard against premature culling before first RMS value is available
  - Same for priorities

Prioritizing

- Sound designer specifies base priority
  - None, Low, Medium, High, Permanent
  - Combines with loudness into final priority at run-time
- Priority determines rendering order
  - When max. rendering time is reached, expel quiet sounds
  - Ensures we should always hear the loudest sounds
    - Use Permanent priority carefully!

Mixing

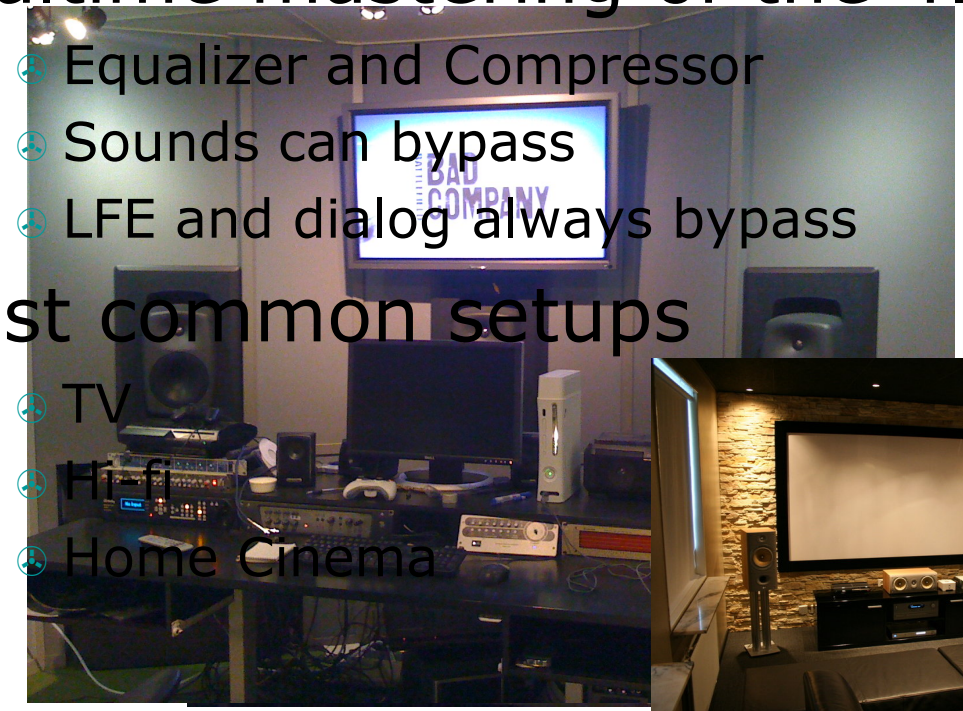
- Traditional scenario
  - Mixed at the end of the project
  - Spent at least a few weeks
  - Did nothing else
- Fundamentally different approach with HDR audio
  - A lot of the complexity ends up being handled by the sound system automatically
  - Sound designer ”only” tweaks loudness and attenuation values

Mixing

- Battlefield: Bad Company
  - Mixed throughout the project
  - Spent at least an equivalent of 6 months
  - In fact, the game was always mixed
    - Ship it!
  - Only a couple of days spent finalizing the mix at the end
    - Including tweaking the listening settings

Mastering

- How do you listen to your game?
  - We can't predict all setups
- Realtime mastering of the 4.0 mix
  - Equalizer and Compressor
  - Sounds can bypass
  - LFE and dialog always bypass
- Most common setups
  - TV
  - Hi-fi
  - Home Cinema



Let’s listen to the result!

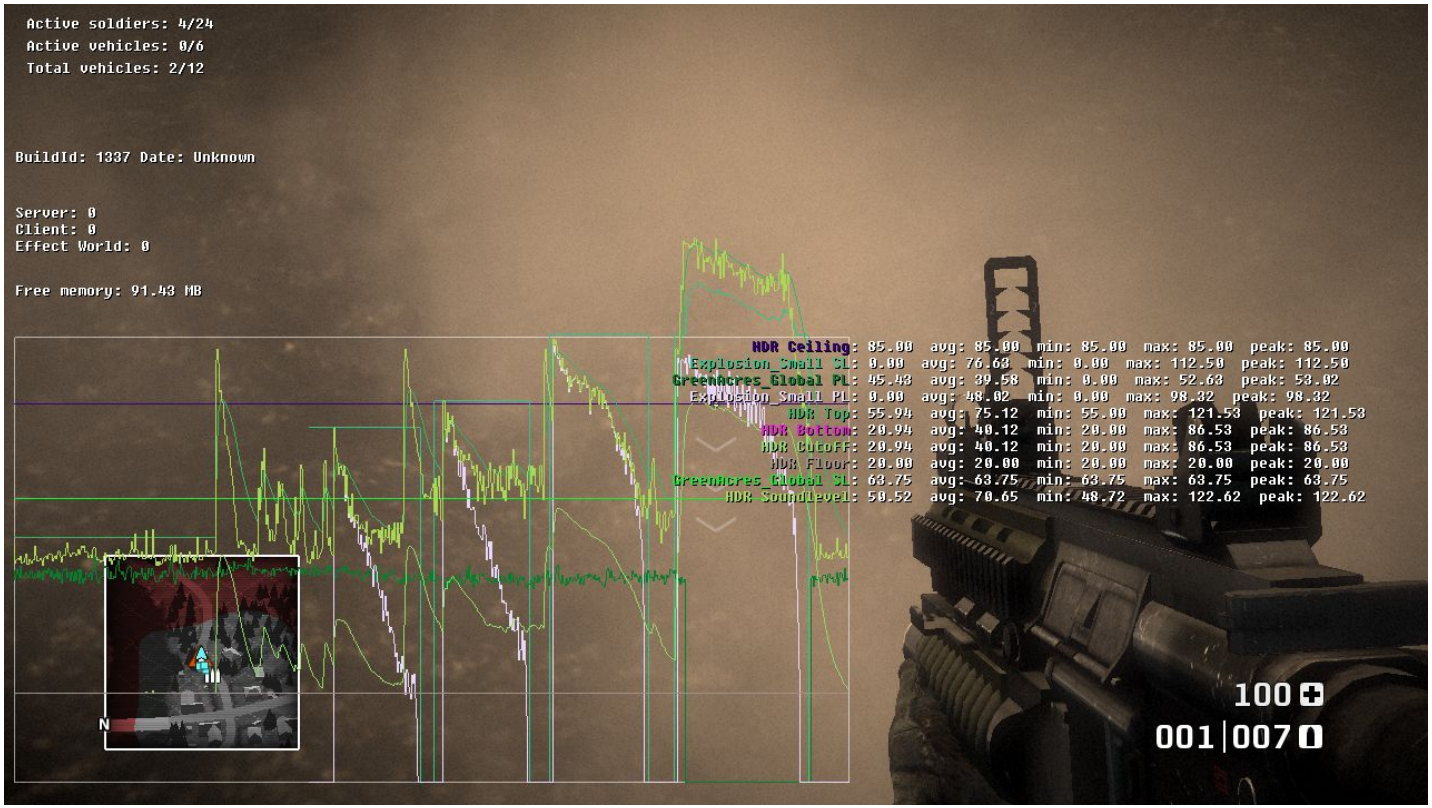
Conclusions

- Suits first/third-person games
  - Most likely other types of games as well
- Battlefield: Bad Company audio on average 92 rated
  - Awesome sound design may have contributed ☺
  - A case of great tech enabling awesome design
- High Dynamic Range audio works!
  - Our gamble definitely paid off
  - The future for HDR audio looks ”loud”!

In the year *2000*

- More accurate perceived loudness
  - Tweak distance model
  - Take frequencies into consideration
  - More fitting normalization of source data
  - Improves culling, prioritization and thus the whole sound scape
- Better workflows and debug info
  - Live-tweaking was crucial, but could be improved
  - Fix things that didn’t get a lot of attention due to focus on shipping Bad Company
- Optimize, Optimize, Optimize!

Questions?



Contact: anders.clerwall@dice.se