

Università di Milano
3D VideoGames 2018/2019




3D Game Audio *(notes)*

Marco Tarini



1

Course Plan



- lec. 1: **Introduction** ●
- lec. 2: **Mathematics** for 3D Games ●●●●●
- lec. 3: **Scene Graph** ●
- lec. 4: Game 3D Physics ●●● + ●●●
- lec. 5: Game Particle Systems ▶
- lec. 6: Game 3D Models ●◀
- lec. 7: Game Textures ●●
- lec. 8: Game 3D Animations ▶●●
- lec. 9: Game 3D Audio ●↻
- lec. 10: **Networking** for 3D Games ●
- lec. 11: **Artificial Intelligence** for 3D Games ●
- lec. 12: Game 3D Rendering Techniques ●●

2

Game Audio: intro



- Fundamental aspect of game-design
 - Impact cannot be overestimated
 - for immersion
 - for emotion
 - for gameplay
 - for story-telling
 - (remember that we don't focus on game-design aspects in this course)
- The main technical aspects of game sound are, however, quite subtle

3

Sound in games: game-design point of view



- Music
 - authored by: Composers
 - *emotional function*
 - Voiceovers
 - authored by: Dialogs writers + Voice actors
 - *narrative (=story-telling) function*
 - Sound effects
 - authored by: Sound Designers / Foley
 - *informative function*
 - Ambient sounds
 - authored by: Sound Designers / Foley
 - *immersive function*
- e.g.:
dialogs (linear / non-linear)
commentary (non-linear)
narration (linear)
- "Sounds makes it *real*
Music makes you *feel*"

4

Sound in games: game-design point of view



- **Sound effects** are mighty **informative**
 - effective way to make things clear to the player.
 - examples:
 - out of ammo:
 - gun just doesn't shoot → wrong key? a bug?
 - gun goes "click" → player gets it
 - doors closes *behind* player in 1st person view
 - sound door-slam effect: let him know!
 - can substitute / abstract animation. Examples:
 - character collects object
 - object just disappears from scene → cheesy
 - pick-up animation? → hard to do right, delay affects gameplay
 - add pick-up sound instead (abstract) → acceptable
 - character changes outfit (RPG)
 - just swap character models → cheesy
 - add cloth undressing+dressing sound (abstract) → acceptable

5

Sounds in games: dev-team roles



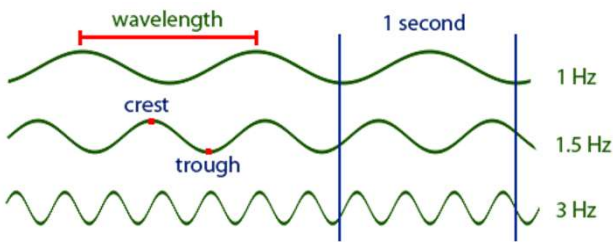
- Composer
- Sound Design
- Foley
- Sound Integrator
- Audio Programmer
- Tool programmer
(for audio related tasks)



6

Sound wave

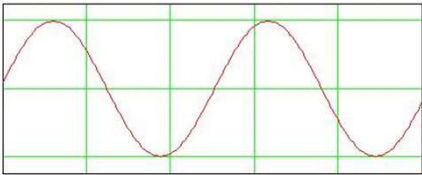
- Air pressure as a function of time
- frequency : (1/sec = 1 Hz)



7

Sound wave

- Air pressure as a function of time
- Waves:
 - frequency (Hz, audible = ~32 to ~16K),
 - amplitude (→ “volume”, level, perceived loudness)



- Perception
 - as with most senses, response is roughly logarithmic with physical quantity (e.g. see: [decibel](#) for [amplitudes](#), [notes](#) for [frequencies](#))

8

Sound wave & perception (a beginner's guide)

What it is:
physical property of the sound wave

What it is perceived like:
by the human hearing system

<p>Amplitude (crest – trough)</p>	<p>Level i.e. loudness (colloquially, Volume) how loud the sound is</p>
<p>Frequency (1/wavelength)</p>	<p>Pitch how high-pitched or low-pitched the sound is [<i>Ita: acuto o grave</i>]</p>

logarithmic

exponential

<p>Spectrum (which frequencies are present)</p>	<p>Timber, tone</p>
--	----------------------------

9

Sound wave as assets

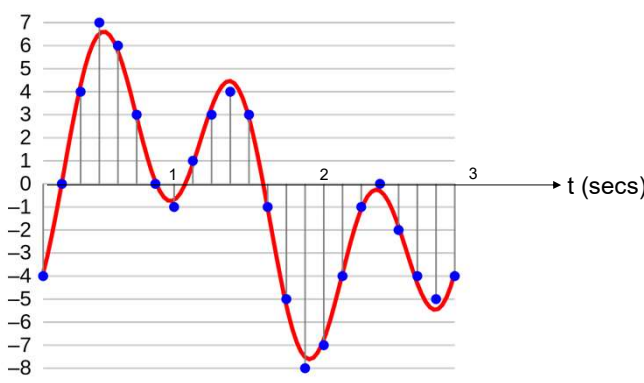
- Air pressure as a function of time

- To digitalize it ("PCM"):
 - **sample** it
 - at some fixed **rate**
 - typically, 24-48 KHz
 - **quantize** samples
 - at some fixed **precision**
 - typically, 14-24 bits per sample
 - then maybe **compress** it

10


PCM – Pulse Code Modulation

- Toy example: 8 Hz sampling, 4 bit quantization:



11

Middleware for sounds in games





Libs: [OpenAL](#) , [Wwise](#) ...

25

Sound Rendering: *basic playback tasks*

in any game,
even in a 2D setting

Main Asset:  the sound **buffer**




the digitalized sound wave,
ready to be sent
to the speaker

- **Mixing**
 - **Linear combinations** of waves
 - E.g. cross fade two sound, maybe with transition functions etc.
- **Tweak / Tune:** (useful to randomize sounds – e.g. footsteps!)
 - **Level** (~“loudness”) – **amplitude scaling**
 - both **pitch** and **speed** – **time scaling**
 - *only pitch*, or *only speed* (a bit less trivial)
- **Sound filters**
 - **convolutions** of sound buffer
 - useful to add **procedural** effects effects as **reverb**, **echos...**
- **prioritization**
 - why: because limited «polyphony»
engine can mix only up to N sounds (e.g. N = 64)
 - solution: game dev assigns a priority to each sound fx

26

Sound Rendering in 3D games

3D (or, “spatialized”) sound

- sounds which are:
 - **emitted** from a virtual source
(somewhere in 3D) ←
 - **received** from a virtual microphone
(somewhere in 3D) ←
 - **propagated** across the 3D scene
- useful abstractions used in games:
 -  the **listener**
 -  the **source(s)** } sitting in nodes of the scene graph!

note:
position
and
orientation

27

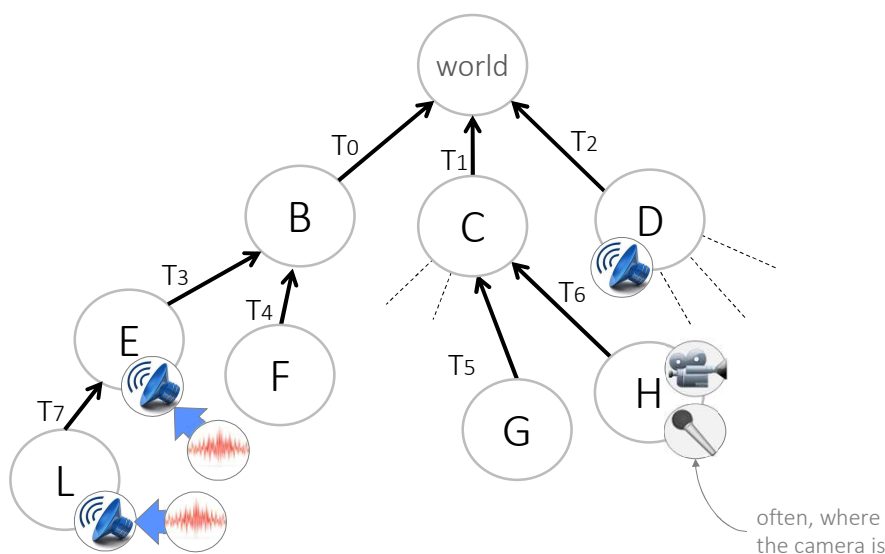
3D (or, “spatialized”) sound: for direct sound propagation



- consequent auto-tuning of
 - **level**: (linked to perceived “loudness”) according to source-listener **distance**
 - with a given (dev-controlled) «roll-off» function
 - E.g. $1/d$ or $1/d^2$
 - **pitch**: (Doppler effect) according to **relative speed** or source w.r.t. listener
 - **interaural time difference (ITD)**: difference of sound arrival time between the two ears. Used by brain for **sound localization**
Gives illusion of sound **relative location** w.r.t. head using stereo speakers. It’s SMALL! e.g. $\sim 10 \mu s$

28

3D (or, “spatialized”) sound



29

Directional (anisotropic) sound emitters

high level (loud)

low level

very low level

$\hat{d}_{emitter}$

direction of emitter (e.g. z-axis, in emitter space)

versor from emitter to listener

- Level = fall-off function of $(\hat{d}_{emitter} \cdot \hat{d})$

30

Why listener orientation is also important

Time L

Time R

Level L

Level R

interaural time difference

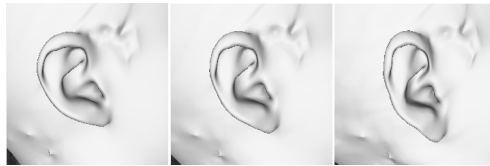
interaural level difference

anisotropic spectral cues

31

Anisotropic sound cues for personalized ear shapes (advanced task!)

- Spectral clues: a “anisotropic” stereo sound filter which depends on sound incoming **direction**
 - in listener reference frame (listener orientation counts!)
- Requires a 3D model of the listener’s hear.

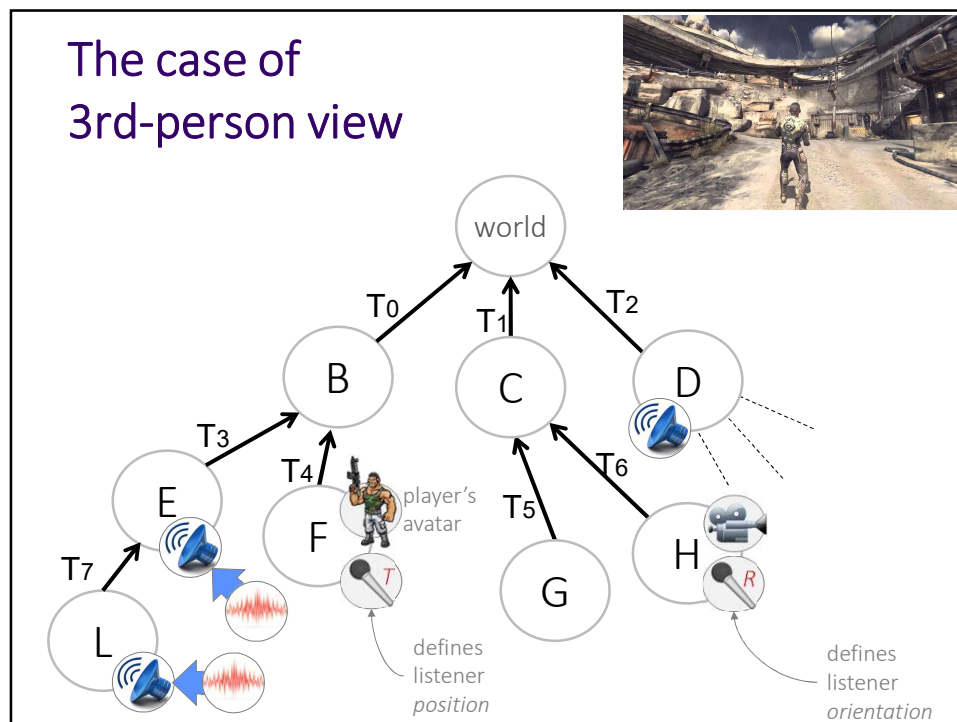


- More commonly, approximations are used

“Reconstructing head models from photographs for individualized 3D-audio processing”
M Dellepiane et al, CGF 27 (7) - (Pacific Graphics)

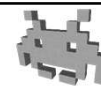
32

The case of 3rd-person view



33

Sound Rendering: sound propagation in the 3D scene



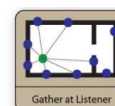
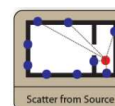
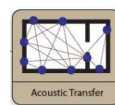
- So far, we only considered the 3D effects of sound direct propagated from emitter to microphone
- Sound waves interact with matter
- Three basic phenomena:
 - Absorption:

34

Sound Rendering: sound propagation in the 3D scene

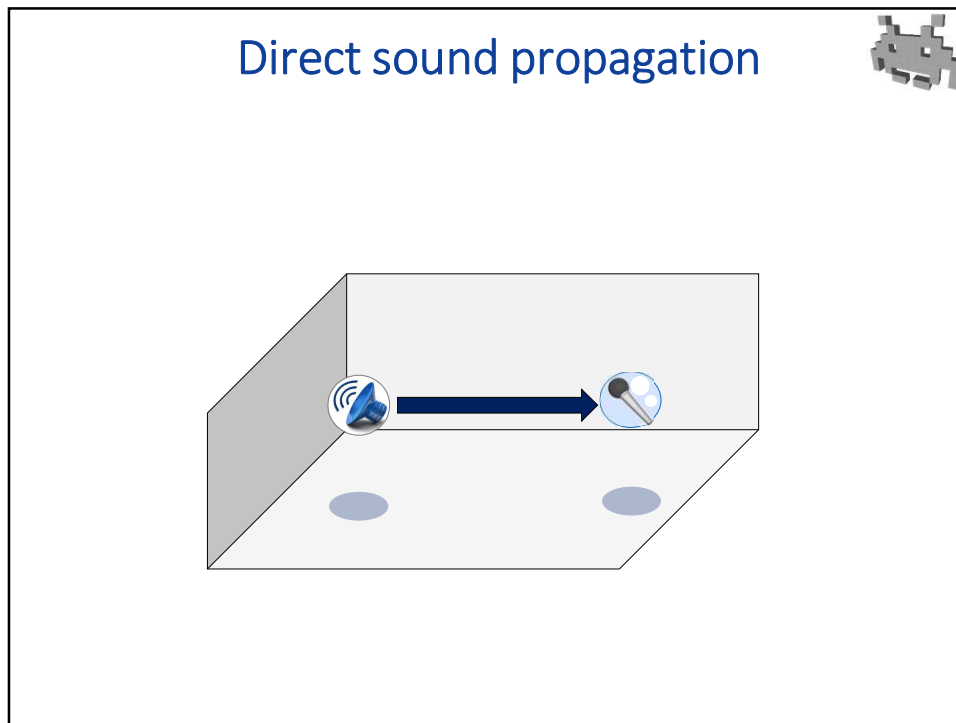


- Reuse **collision proxies!**
- Targets simulation of effects like:
 - muffling / absorption
 - occlusion, obstruction
 - reverb / echoes
- Research topic
 - Currently: no standard solution adopted by 3D games
 - Often, tricks coded *ad-hoc* by the **sound programmer**

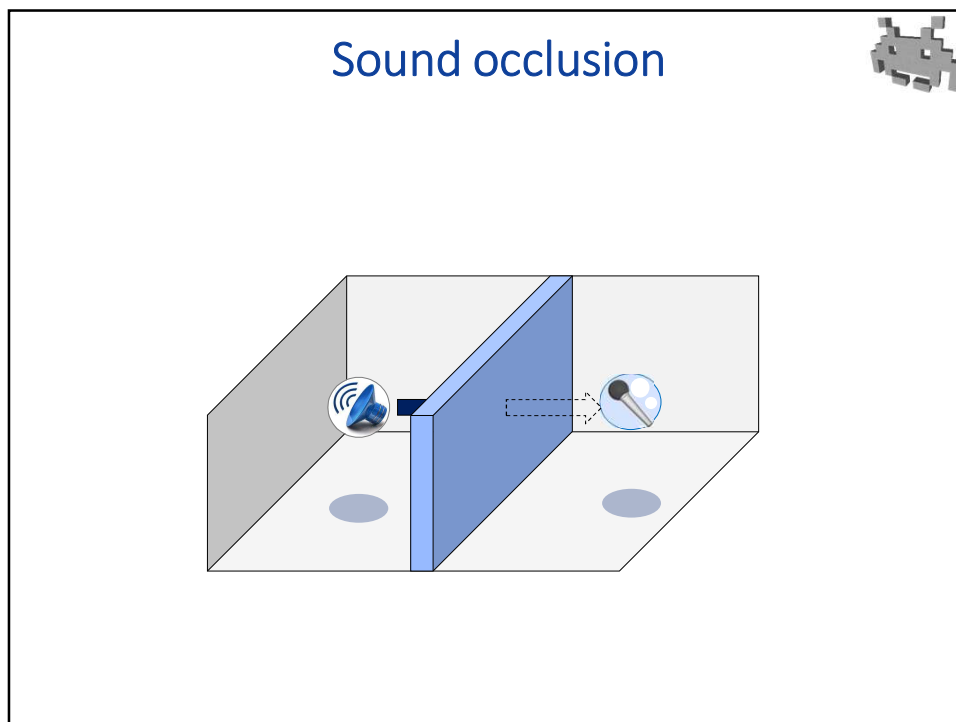


E.g. see: "Interactive Sound Propagation using Compact Acoustic Transfer Operators"
Lakulish Antani, Anish Chandak, Lauri Savioja, Dinesh Manocha
SIGGRAPH 2012

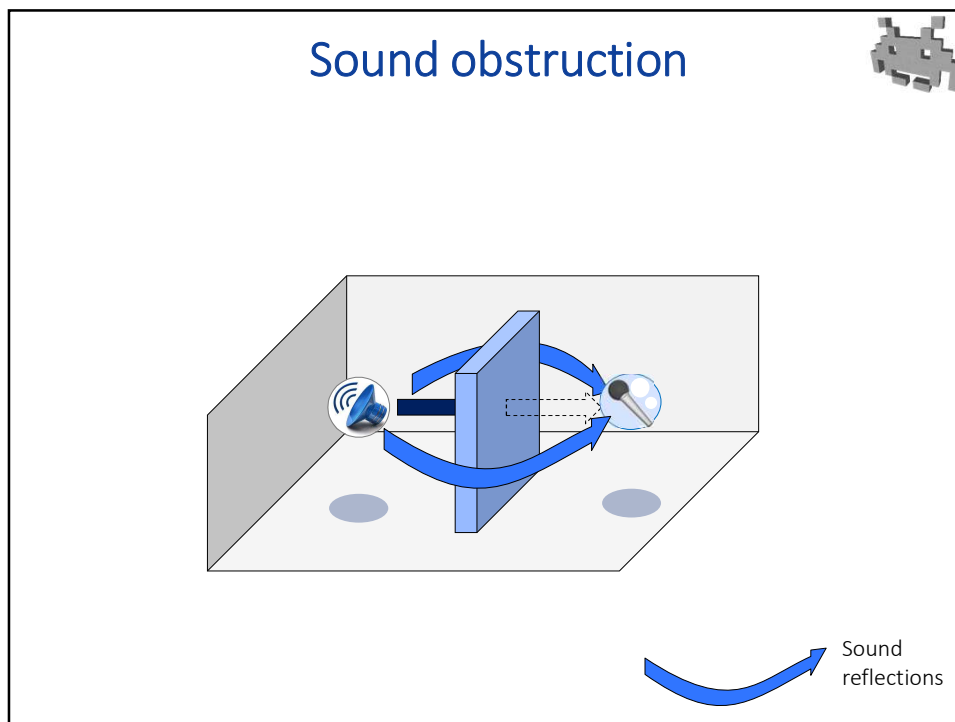
35



36



37



38

Sound Rendering: full computation of sound propagation in scene

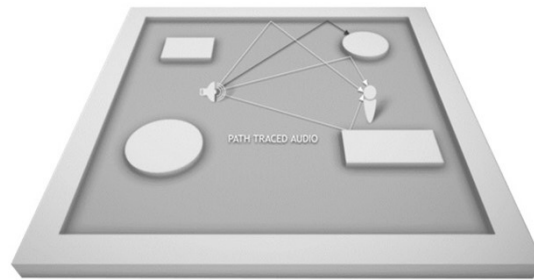
- e.g. for collisions
- using physical material specification
- not (yet?) used in games
 - but active research topic

E.g. see: "Toward Wave-based Sound Synthesis for Computer Animation"
Jui-Hsien Wang, Ante Qu, Timothy R. Langlois, Doug L. James
SIGGRAPH 2018

39

Sound reverb

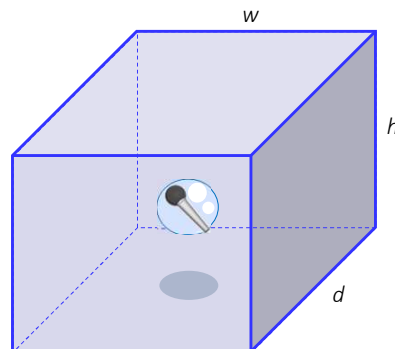
- Solution 1: path tracing (expensive!)



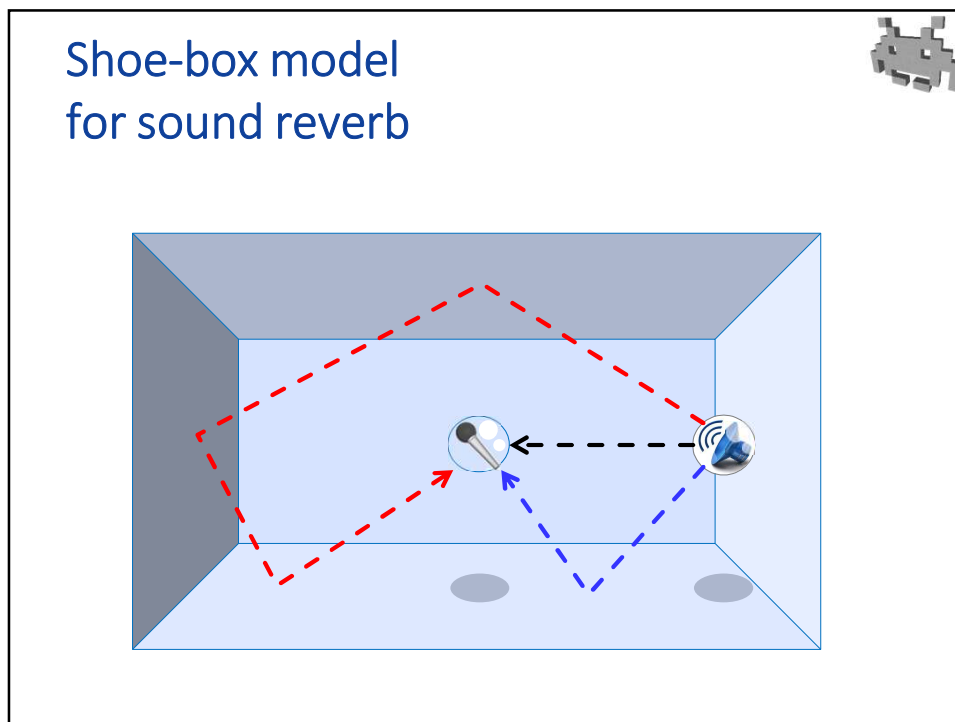
40

Sound Rerverb

- Solution 2: «shoe-box model»
 - An approximation with closed-form formulas



41



42

What triggers sound fxs in a typical game-engine?

- fxs explicitly started from **scripts**
 - e.g. at **collision response**
 - e.g. accompanying all sorts of **game logic**
 - anything from “doors opening” to “level completed”
- fxs associated to scene **Objects**
 - constantly looped fx from a source, e.g. a radio
- fxs associated to **interface elements**
- fxs as **Actions** of the **AI** (see AI lecture)
 - see: **AI** for **NPCs**
- fxs associated to **Animations** (see animation lecture)
 - e.g. *footsteps* fxs during walk
 - e.g. *detach from ground / Land* fxs during jumps
 - e.g. *air-swishes* during sword swings
 - convenient to ease action/sound synchronization

43

Authoring sound effects (task of the Sound Designer)



- Remember: as any asset, you can buy / get them from **Libraries / Repositories**
 - Common (so many needed fxs, so little time)
- **Capture**
 - Digital artist: “Foley”
 - Field capture (for ambient sounds → drones)
- **Synthetize**
 - by sound editing
 - (rarer)

44

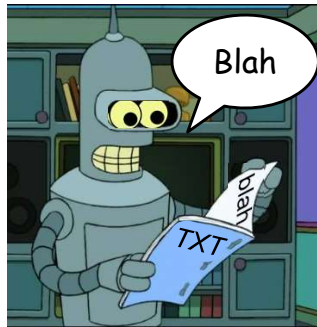
Voice Overs



- Two kinds:
 - Linear
 - e.g. cutscenes, narrations
 - Non linear (e.g. state machine – see AI lecture)
 - e.g. multiple answers dialogs
 - e.g. running commentary (of a football match)
- Technically, it’s nothing special. Just a sound fx.
- But, several practical challenges:
 - Lots of assets! (file names, folders nightmare)
 - Localization often needed
 - Expensive production (\$\$\$)
 - During early stages: better to use placeholders!

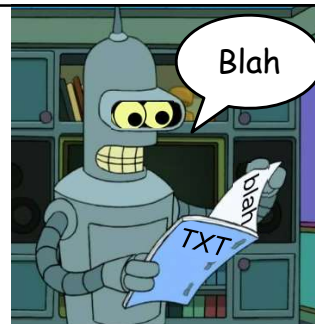
45

Speech Synthesis (or “text to speech”)



46

Speech Synthesis (or “text to speech”)



- A.I. frontier
- currently: still not good enough
 - not *believable* enough
 - human voice = we are all expert = difficult to trick us
 - audio “uncanny valley” ?
 - not *expressive* enough (emotions, characterizations)
 - i.e., virtual voice actors are not ... good voice actors
- just a matter of time?
- when it will be here, it will
 - free games from most issues of **voice-over assets**
 - get us all the usual advantages of **procedurality**

47

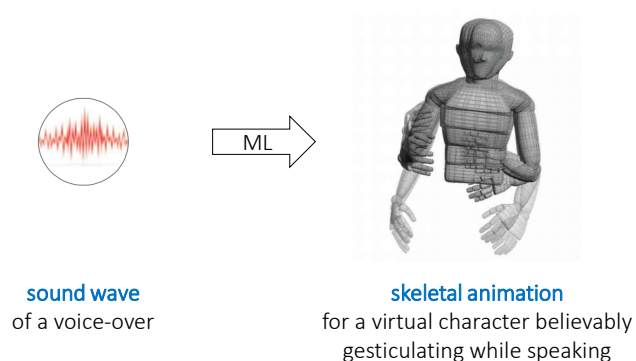
A summary of authoring sound assets

- Synthesized / simulated / procedural fxs :
 - baked
 - (rare)
 - Captured fxs :
 - hardware: a good microphone!
 - by: "Foley artists"
 - very often: just bought / downloaded from repositories
 - Voice :
 - hardware: a good microphone!
 - by voice actors
 - (sometimes, during motion capture sections)
 - speech synthesis? won't be used (for some time yet)
 - Composed (for music) :
 - musicians: frequent 3rd members of 3-man dev teams
 - recent improvements of tools (both HW and SW)
 - e.g. chorus with arbitrary lyrics now attainable
 - a few game composer gained substantial fame!
- } then, sound editing

48

Research topic: from voiceovers to NPC animations

- With Machine Learning (data driven)



"Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows"
Simon Alexanderson et al, CGF (Eurographics 2020)

49