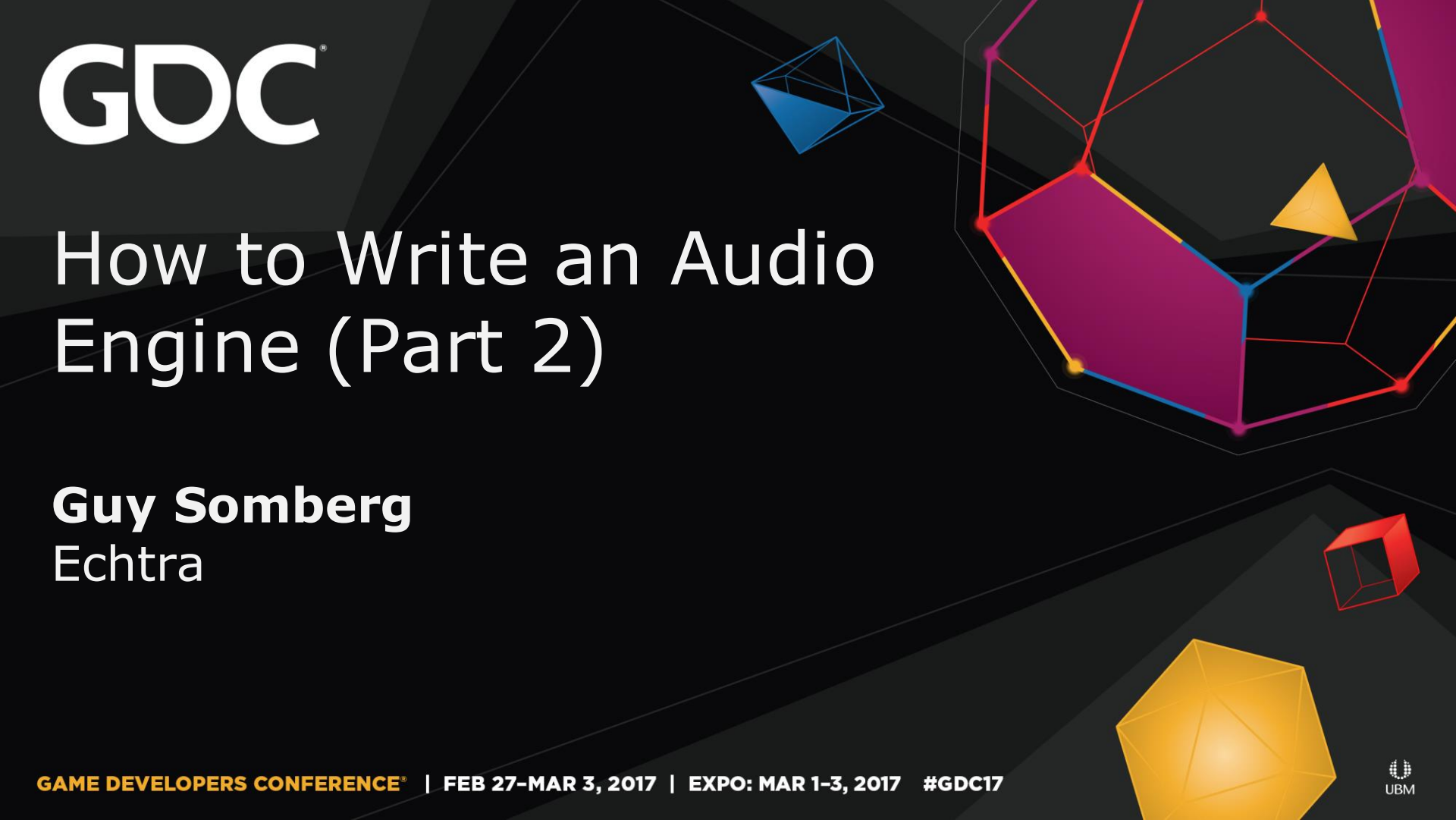


How to Write an Audio Engine (Part 2)

Guy Somberg
Echtra

GDC2015

- How to Write an Audio Engine
  - Part 1, apparently
  - Available in the GDC Vault
- It’s pretty good! You should watch it.

GDC2015

- Sound Engine State Machine
- 3D Listener
- Multithreading
- Mixing
- Reverbs and Ambience
- Platform-specific stuff

GDC2017

Differences

GDC2015	GDC2017
99% Light Speed	90% Light Speed
151 Slides	95 Slides
Lots of code	Minimal code
8 State Machine Slides	3 State Machine Slides
100% Awesome	100% Awesome

Order of Operations

- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

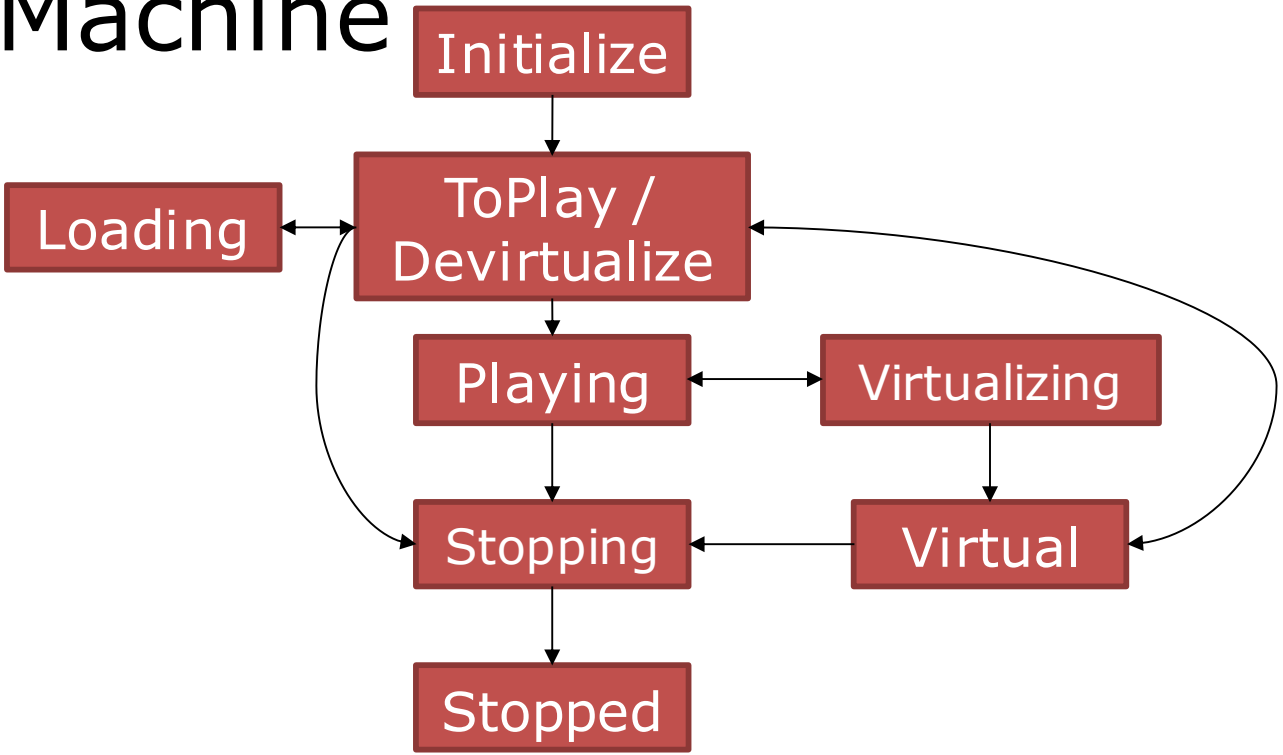
First, a correction

- MacOS/iOS have no `sem_timedwait()`
  - True only for older API levels
  - At Telltale Games, we were targeting older MacOS and iOS builds, so we were limited
  - If you're targeting a newer version, then move on, nothing to see here…

Order of Operations

- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

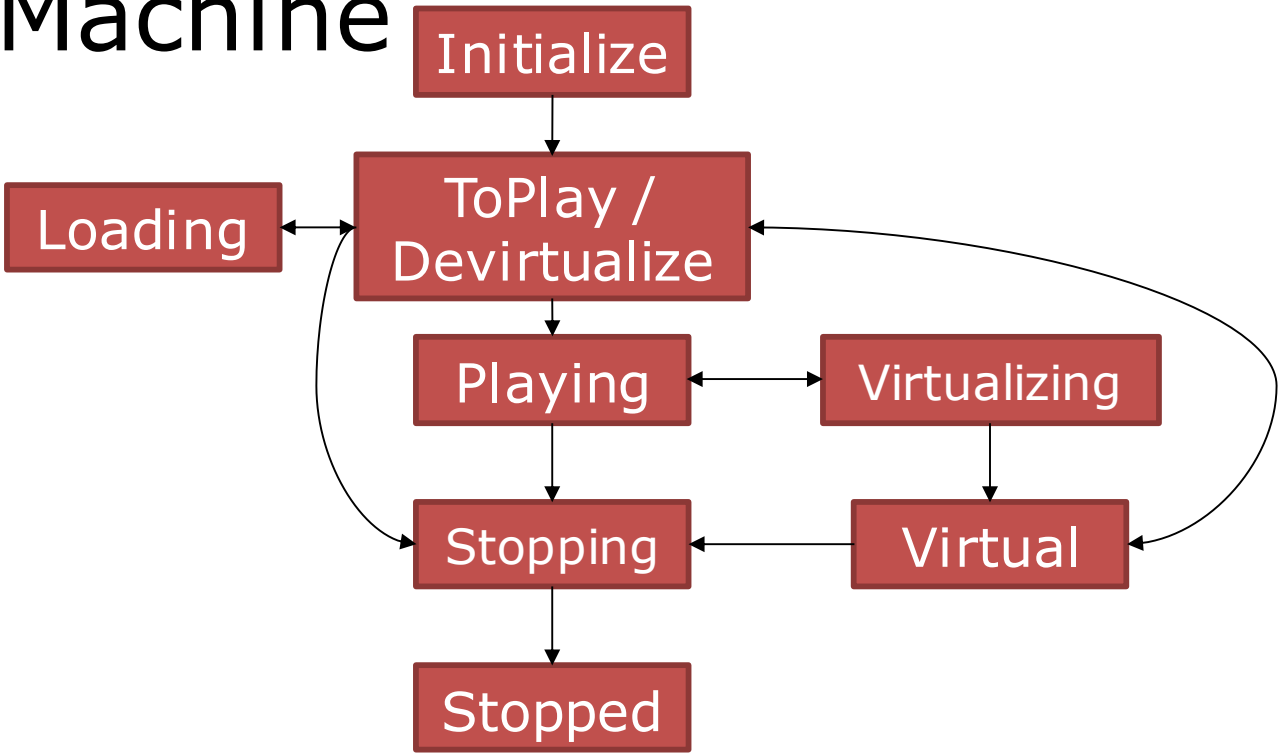
State Machine



Games That Have Shipped with (Some Variation of) This State Machine

Hellgate: London	(1st/3rd Action RPG)
Mythos	(Dungeon Crawler)
Bioshock 2	(FPS)
The Sims 4	(…The Sims?)
Tales From the Borderlands	(Telltale Adventure)
Game of Thrones	(Telltale Adventure)
Unannounced Echtra Title	(Secret)

State Machine



Order of Operations

- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

What is a Virtual Sound?

- A sound that’s not important enough to hear
- For the purpose of controlling:
  - Performance
  - The mix

Examples

- Small fires spread throughout your large level
- A bullet impact a mile away
- The third enemy firing a machine gun from the same guard tower
- Ambiences (while scoped in with a sniper rifle)
- Footsteps (while firing your machine gun)

How to virtualize a sound

- One-shots:
  - Don’t even bother playing the sound
  - Check only during ToPlay
- Looped sounds:
  - Fade over 0.25-0.5 seconds
  - Stop the channel
  - Check regularly

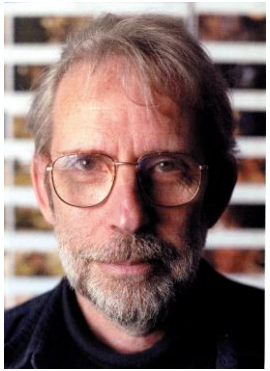
Virtualization Checks

- Must be cheap/fast
- Must be configurable
- Must be short-circuitable

My Common Three Checks

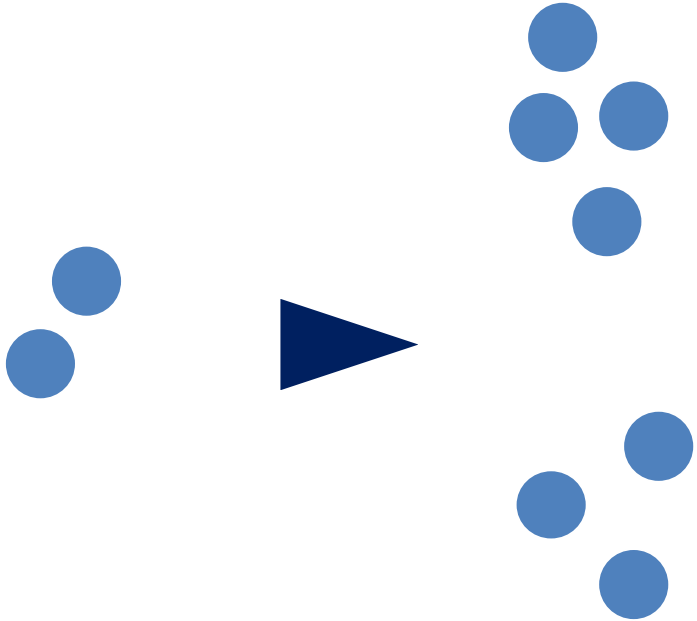
- Out of Range
- Volume is sufficiently close to silence
- Max Within Radius

Walter Murch

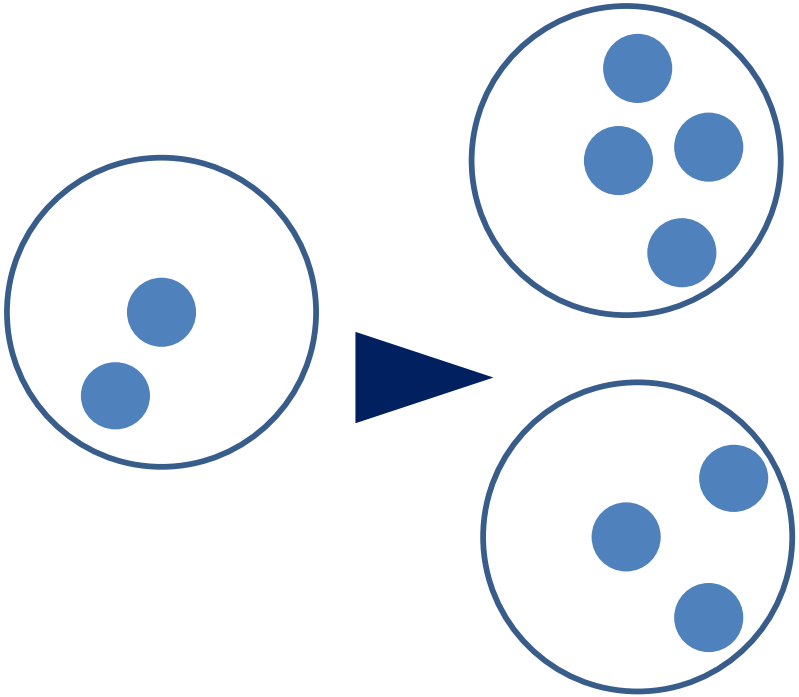


- http://transom.org/2005/walter-murch/
- Law of Two-and-a-Half
- "…if there was one robot, his footsteps had to be in sync; if there were two robots, also, their footsteps had to be in sync; but if there were three robots, *nothing* had to be in sync. Or rather, any sync point was as good as any other!"

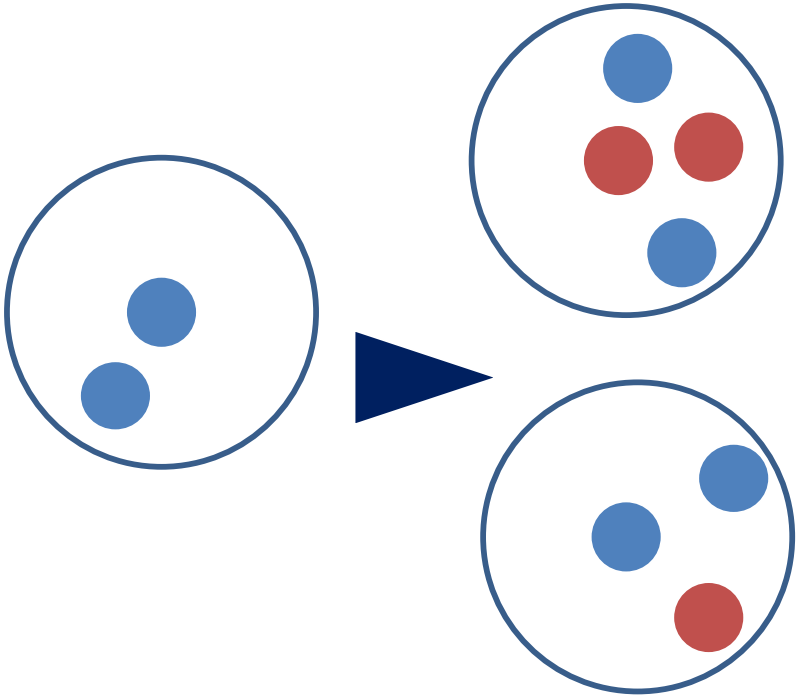
Applying the Rule



Applying the Rule



Applying the Rule



Making the Rule Work

- Performance
  - Iterate only over matching sounds
- Configuration:
  - Each sound has “Max Within Radius” and “Radius”
  - Greater flexibility: “MWR Group”

Virtualization

- Quick to implement
- Extensible
- Effective way to clean up the mix

Order of Operations

- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

Ever Had This Experience?

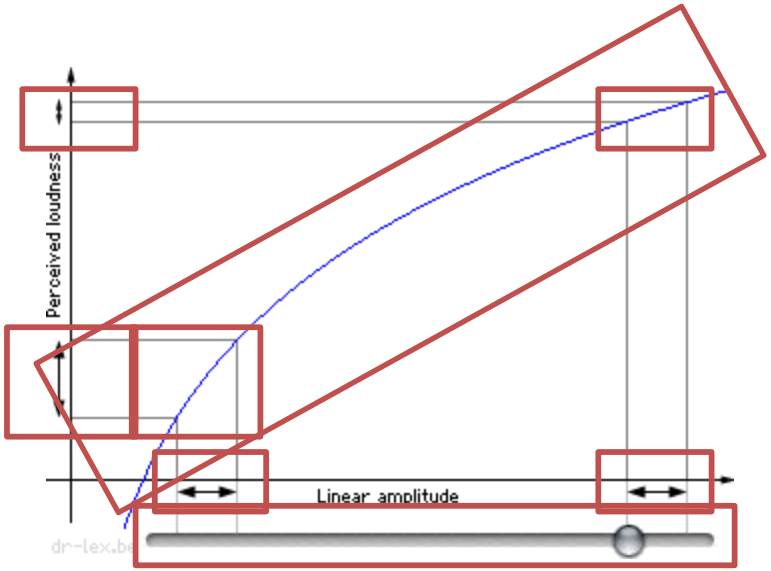
Settings Menu

Live Demonstration

Why Does This Happen?

- The volume slider is linearly interpolating from 0..1
- But sound is logarithmic!

What is going wrong?



http://www.dr-lex.be/info-stuff/volumecontrols.html

How to Fix It

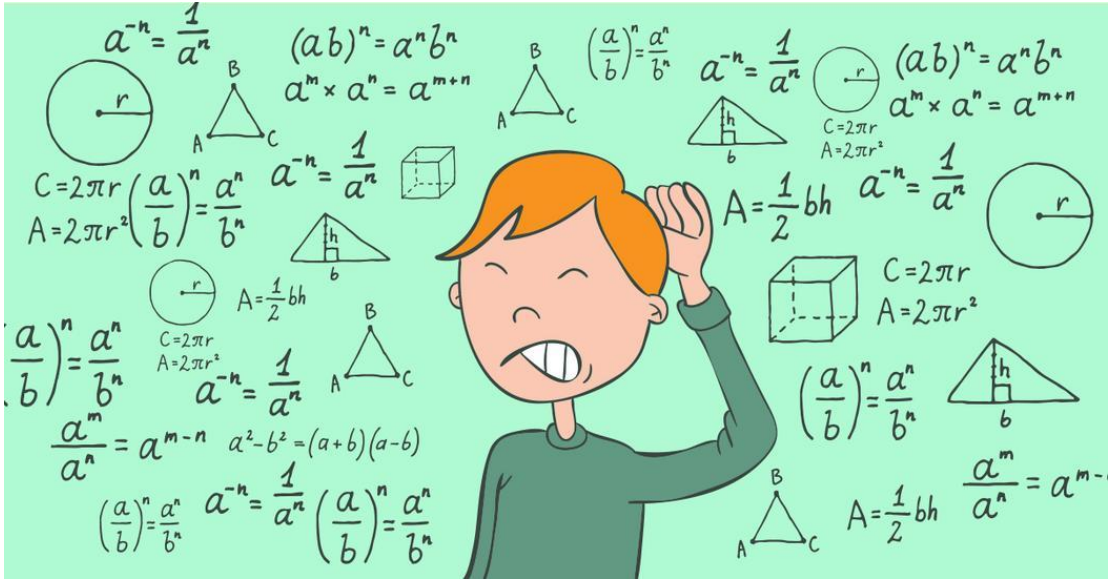
- Instead of being linear, the slider should be *exponential*
- Because *log(exp(x)) == x*

Enough Hand-Waving!

- We have y = a * exp(b*x)
- At x = 1, y = 0dB
- At x = 0, y = (Your dynamic range)

- 40dB worked well for me

Math!


$a^{-n} = \frac{1}{a^n}$
$(ab)^n = a^n b^n$
$a^m \times a^n = a^{m+n}$
$C = 2\pi r$
$A = 2\pi r^2$
$\left(\frac{a}{b}\right)^n = \frac{a^n}{b^n}$
$A = \frac{1}{2}bh$
$\frac{a^m}{a^n} = a^{m-n}$
$a^2 - b^2 = (a+b)(a-b)$

The Pink Box at the End of the Chapter

- y = a * exp(b*x)
- d = chosen dynamic range
- a = 1/(10^(d/20))
- b = ln(10^(d/20))

Show Me the Code!

```
const float dynamic_range = 40.0f;
const float power = powf(10.0f,
                         dynamic_range / 20.0f);
const float a = 1.0f / power;
const float b = logf(power);

volume = a * expf(slider_value * b);
```

Live Demonstration

Better Yet…

- Do this in your volume sliders, but…
- Mix your game properly, so that the players don’t have to fiddle with the volume
- There is an argument for not having volume sliders at all!

Order of Operations

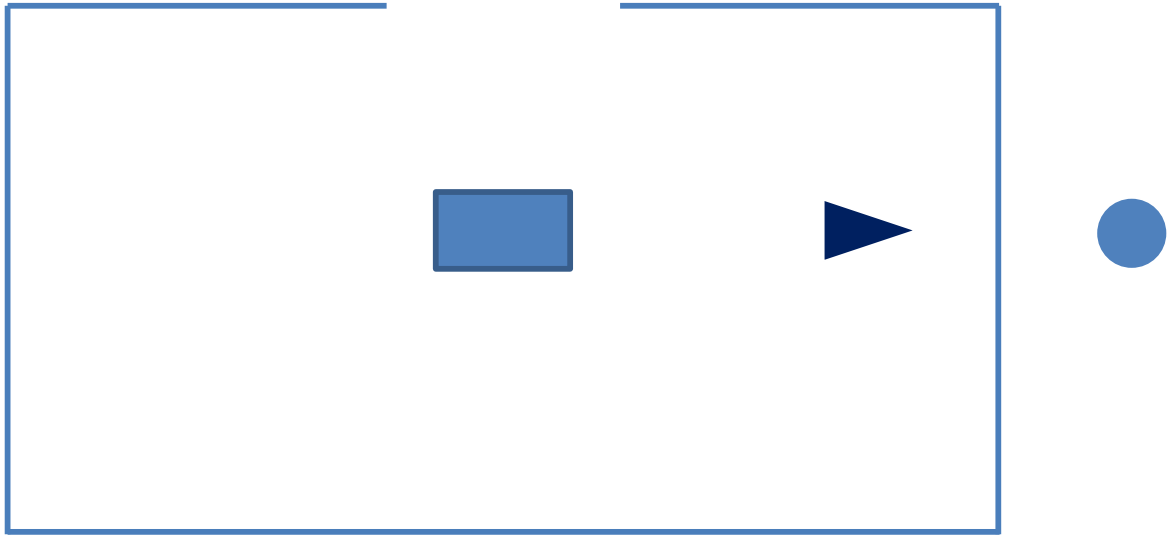
- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

First, the basics

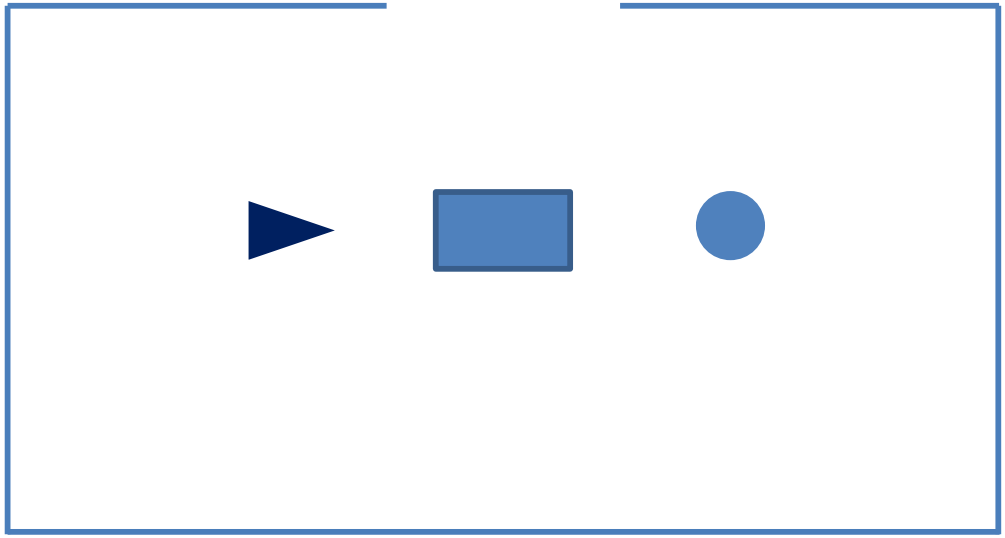
- Obstruction
- Occlusion
- Exclusion

- Colloquially: Obstruction/Occlusion, or just Occlusion

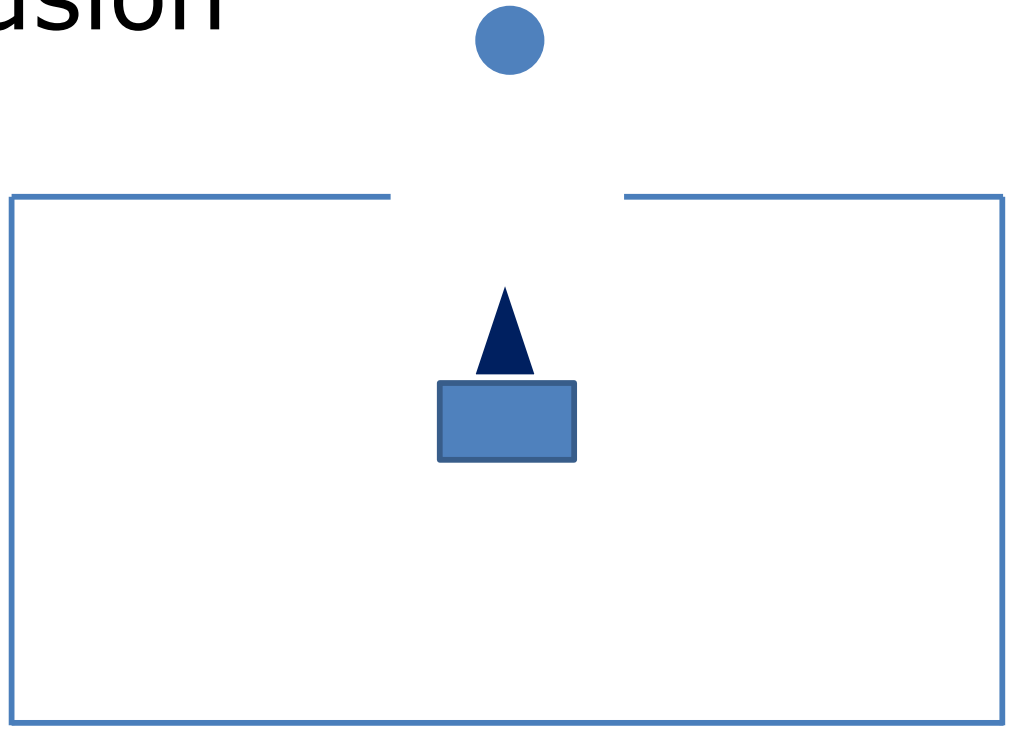
Occlusion



Obstruction



Exclusion



Occlusion Summary

- Occlusion:
  - Direct path obstructed
  - Indirect path obstructed
- Obstruction:
  - Direct path obstructed
  - Indirect path unobstructed
- Exclusion
  - Direct path unobstructed
  - Indirect path obstructed
- No Occlusion
  - Direct path unobstructed
  - Indirect path unobstructed

Truth Table

		Direct Path	
		Occluded	**Unoccluded**
Indirect Path	**Occluded**	Occlusion	Exclusion
	Unoccluded	Obstruction	No Occlusion

Mapping to Actual Effect

- Direct Path
- Indirect Path

Mapping to Actual Effect

- Direct Path = Volume/Low Pass
- Indirect Path

Mapping to Actual Effect

- Direct Path = Volume/Low Pass
- Indirect Path = Reverb Send

Actual Effects

- Occlusion:
  - Volume reduced
  - Low Pass Filter
  - Reverb send reduced
- Obstruction:
  - Volume reduced
  - Low Pass Filter
- Exclusion
  - Reverb send reduced

Finding Direct/Indirect Path

- Option 1: Do nothing
- Option 2: Raycasts
  - Variation 1: 1 Ray
  - Variation 2: 5 Rays
- Option 3: Pathfinding

Finding Direct/Indirect Path

- Option 1: Do nothing
- Option 2: Raycasts
  - Variation 1: 1 Ray
  - Variation 2: 5 Rays
- Option 3: Pathfinding

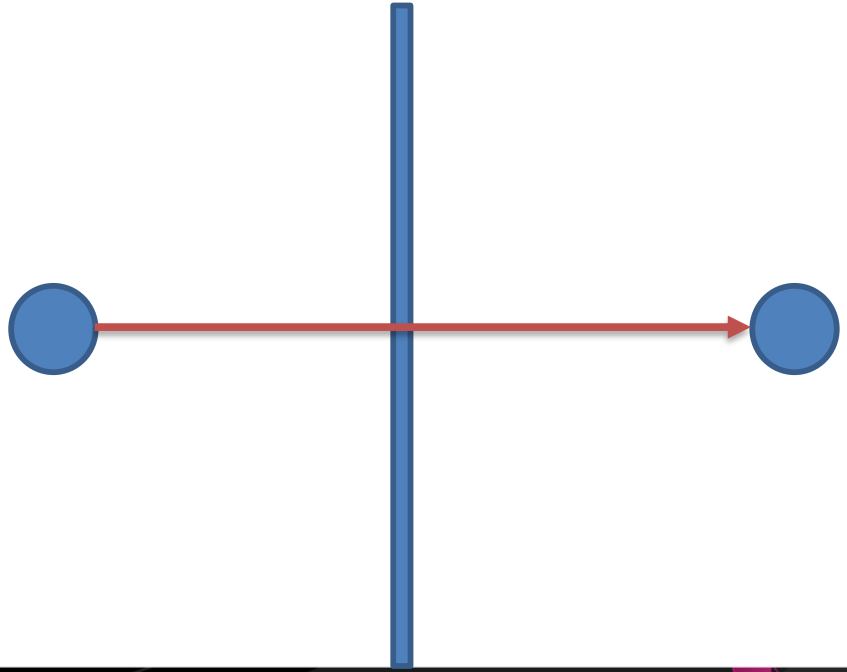
Option 1: Do Nothing

- Advantages:
  - Cheap!
  - Easy!
- Disadvantages:
  - No audible effect

Finding Direct/Indirect Path

- Option 1: Do nothing
- Option 2: Raycasts
  - Variation 1: 1 Ray
  - Variation 2: 5 Rays
- Option 3: Pathfinding

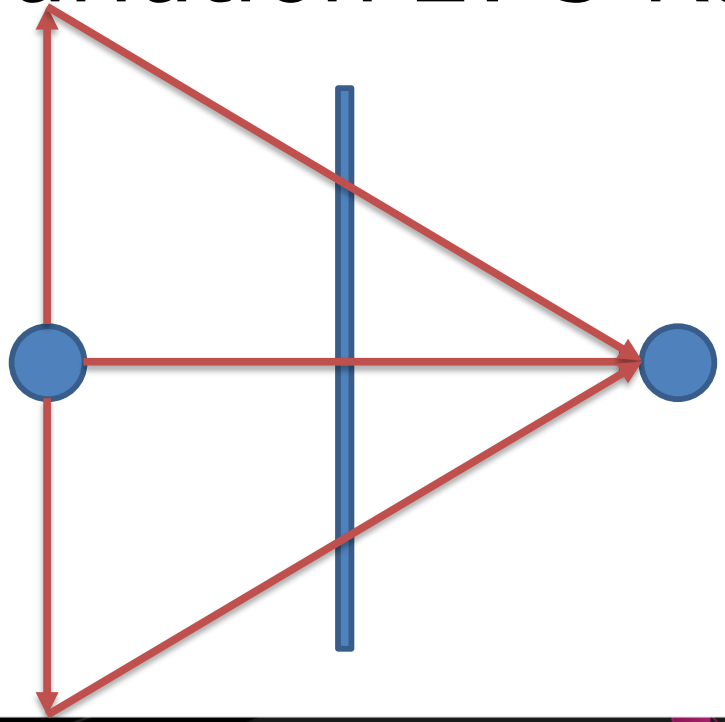
Variation 1: 1 Ray



Raycast Variation 1: 1 Ray

- Advantages:
  - Simple to implement
  - Relatively cheap
- Disadvantages:
  - Only includes direct path information
  - Indirect path must be simulated

Raycast Variation 2: 5 Rays



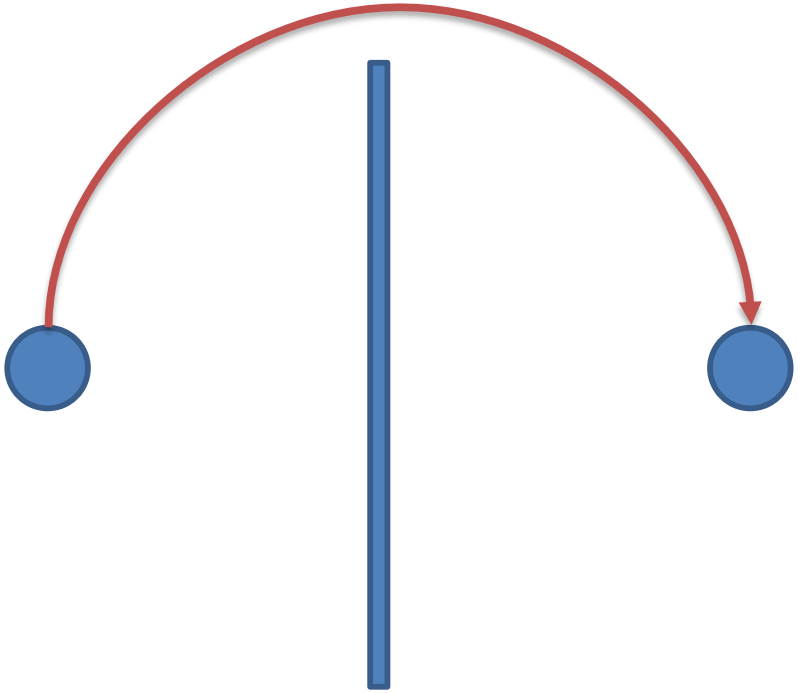
Raycast Variation 2: 5 Rays

- Advantages:
  - Still relatively simple to implement
  - Still relatively cheap
  - Discovers indirect path to area around listener
- Disadvantages:
  - Even more raycasts
  - Coarse indirect path discovery

Finding Direct/Indirect Path

- Option 1: Do nothing
- Option 2: Raycasts
  - Variation 1: 1 Ray
  - Variation 2: 5 Rays
- Option 3: Pathfinding

Pathfinding



Pathfinding

- Advantages:
  - No rays!
  - Path itself provides information on direct and indirect occlusion
  - Path length can be used for attenuation
- Disadvantages:
  - Pathfinding may be more expensive than rays
  - Information on direct/indirect occlusion must be inferred from path

Verdict

- No option is necessarily better than the others
- Choose the option that is appropriate for your game
  - Performance
  - Memory
  - Audible Effect

Order of Operations

- Sound Engine State Machine Reprise
- Virtual Sounds
- Volume Sliders
- Obstruction/Occlusion
- Footsteps

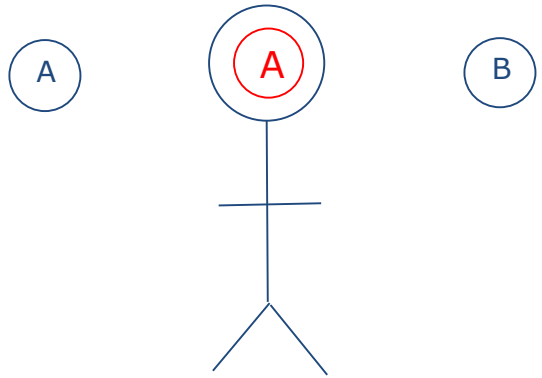

threadneck

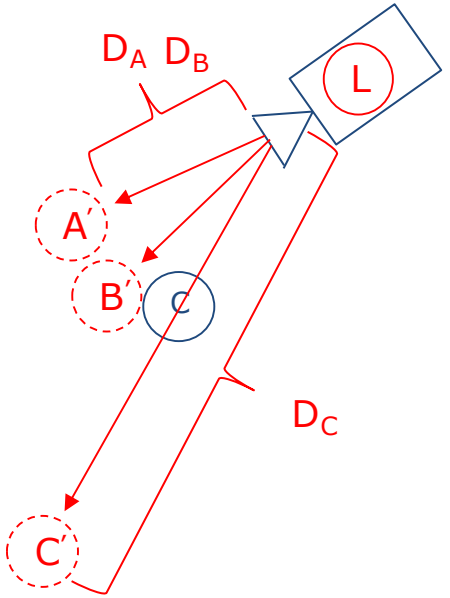
Footsteps

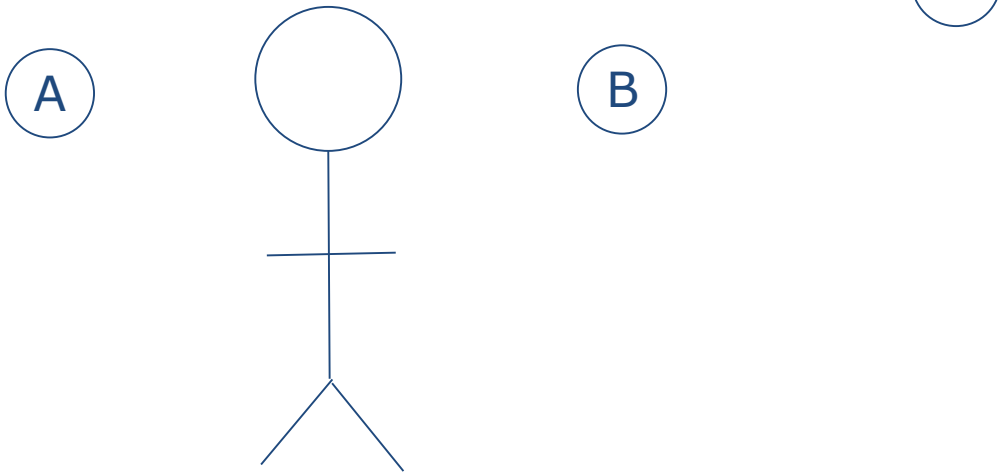
- Attenuation model exception
- How to detect footsteps
- Footstep materials
- Footstep volume
- Variations

Remember this?

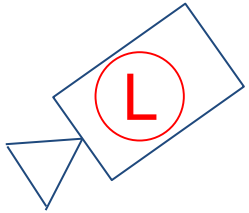
- No?
- Okay, quick recap…

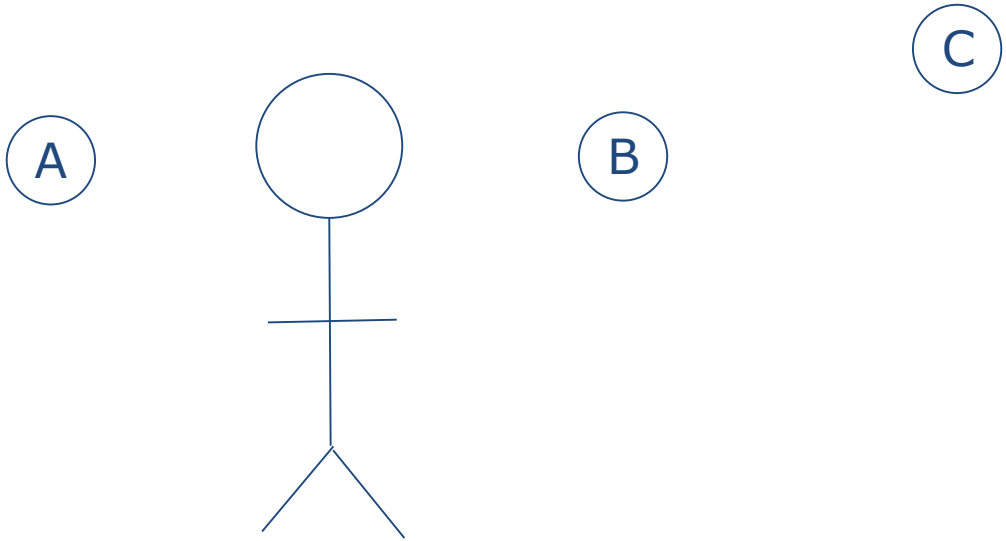




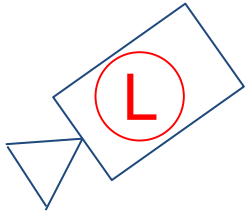

A
B
C

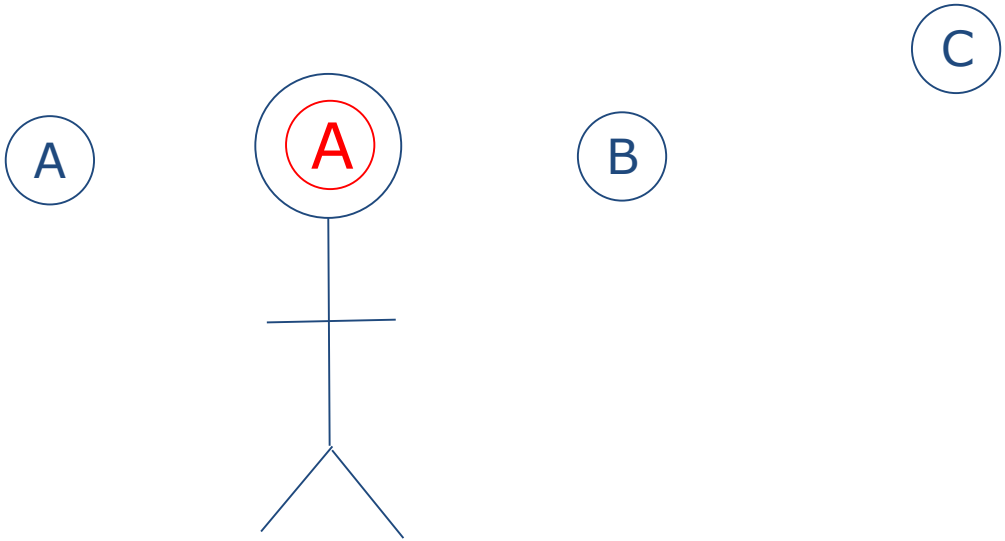
Put the Listener in the Camera



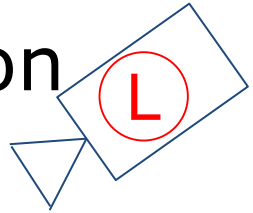


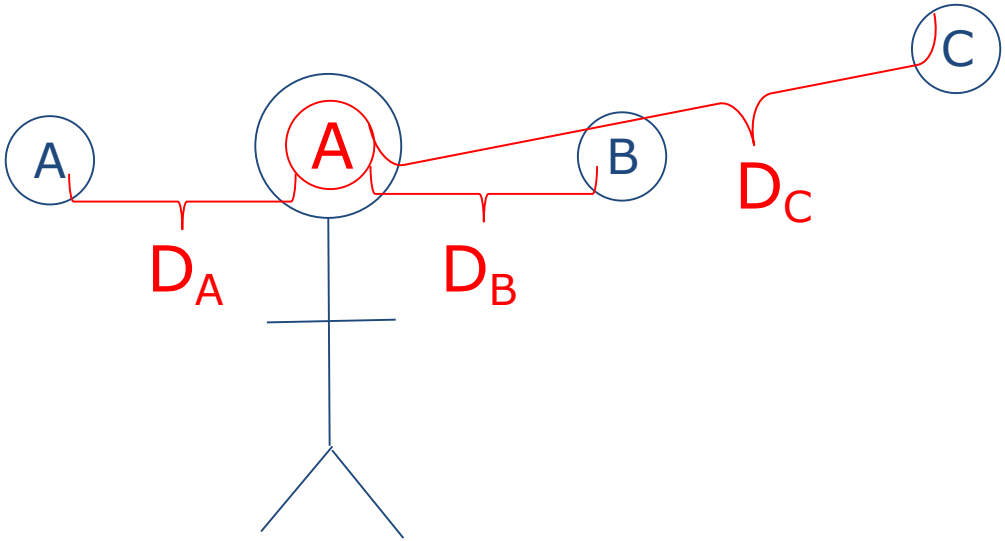
Select an Attenuation Position



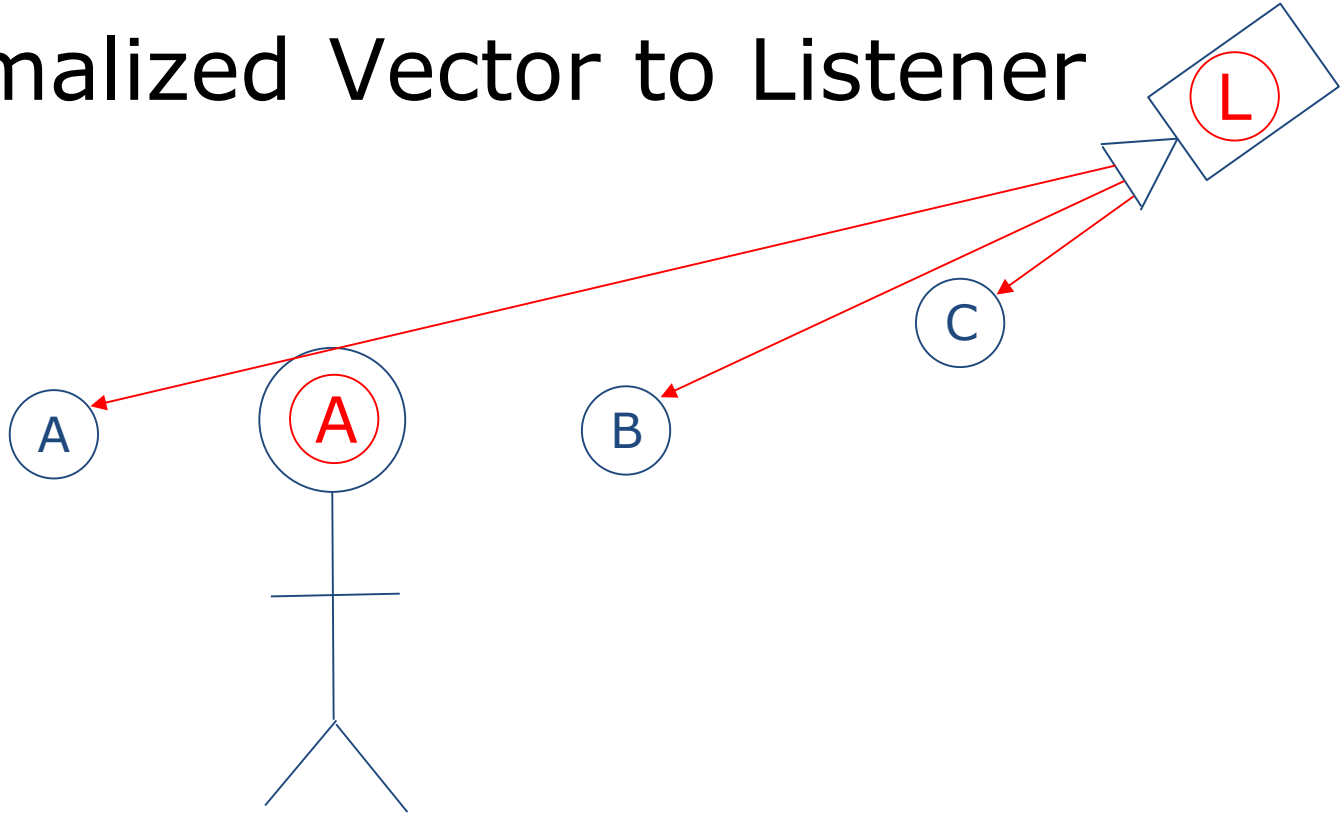


Measure Distance to Attenuation Position

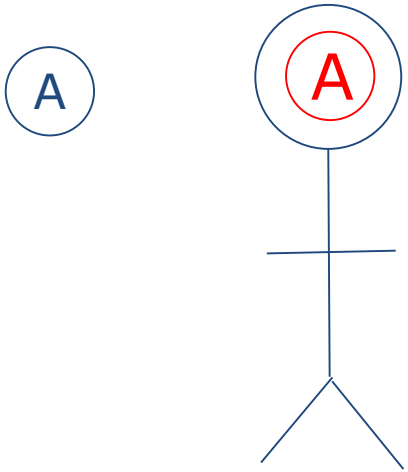


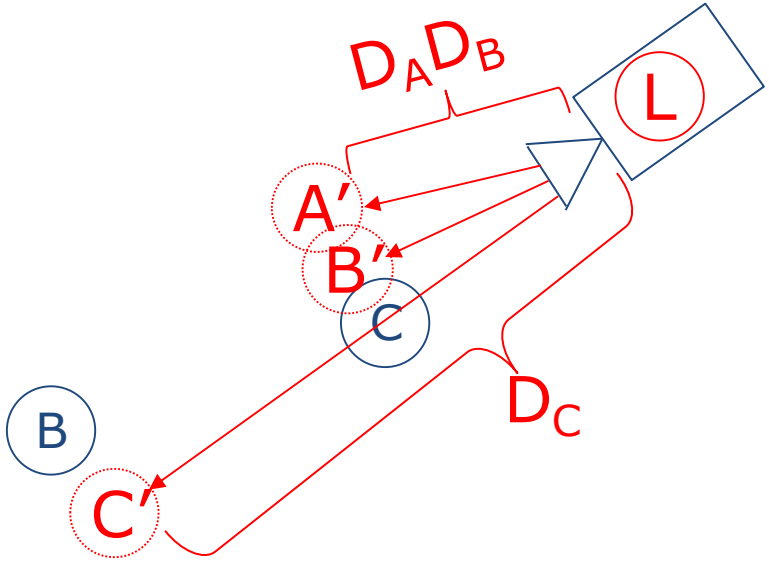


Find Normalized Vector to Listener

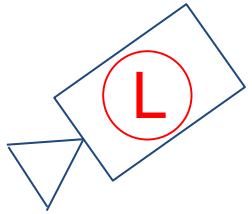


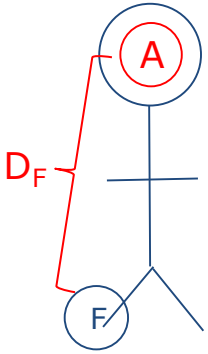
Reposition Sounds



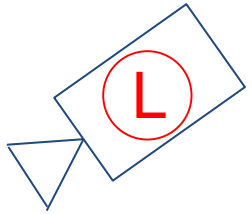


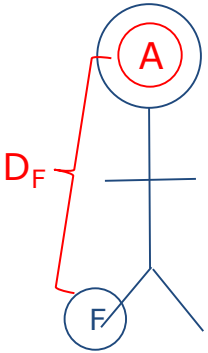
Footsteps



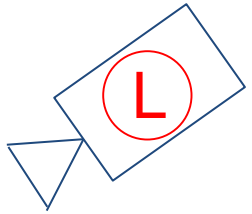


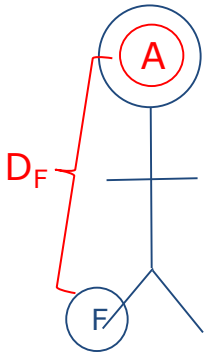
Footsteps





Footsteps





Footsteps

- Attenuation model exception
- How to detect footsteps
- Footstep materials
- Footstep volume
- Variations

Detecting Footsteps

- Option 1: Animation Trigger
- Option 2: Location Trigger
- Option 3: Collision Trigger

Animation Trigger

- Designer tags animation with footstep events
- Easy enough
- But: blending multiple animations can be problematic

Blending Problems

33% and 66%

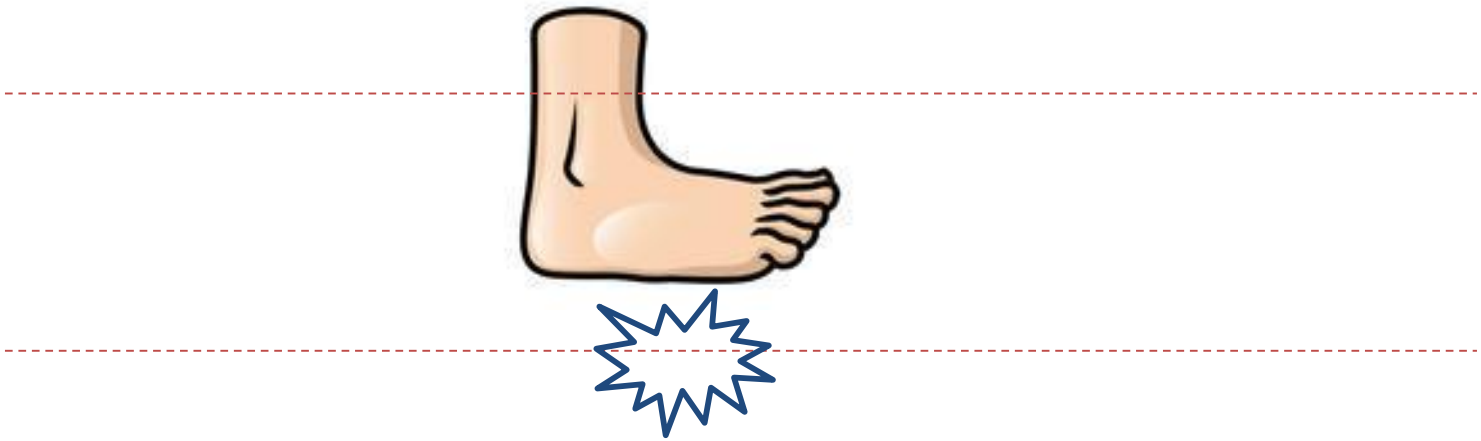
25% and 75%

What do you do when blending walk and sidestep?

Location Trigger

- Trigger when bone reaches minimum point in local space after going above maximum
- Works with blending!

Location Trigger



But…

- Position checks every frame
- Can miss a minimum due to framerate
- Does not detect collision with ground

Collision Trigger

- Attach collision shape to bone and trigger whenever it collides with the ground
- Similar to location trigger, but event-driven
- Works with blending
- But requires care

Footsteps

- Attenuation model exception
- How to detect footsteps
- Footstep materials
- Footstep volume
- Variations

Footstep Materials

- Find material by:
  - Doing a short raycast downwards, or…
  - Swap materials based on entering/leaving zones
- Limit material set:
  - Is water really that different from gore?
  - Is concrete different from tile?

Material Fallbacks

- Default sound per character
- Fallbacks per material type
  - e.g. Gore falls back to water, concrete falls back to tile…

Footsteps

- Attenuation model exception
- How to detect footsteps
- Footstep materials
- Footstep volume
- Variations

Footstep Volume

David Steinwedel:

- Are footsteps important to your game?
- Will anybody miss them if they're not there?
- If you answered No to both, then don't bother with footsteps at all.

Footstep Volume

- Player footsteps are important when the player starts to walk, less important 10 seconds later
- Use mixing tools to describe the desired behavior

Footsteps

- Attenuation model exception
- How to detect footsteps
- Footstep materials
- Footstep volume
- Variations

Variations

- Lots of source sounds
  - Don’t forget heel/toe vs. toe/heel
- Volume/Pitch variation
- Lowpass Filter

Variations

- Be extreme in randomization
- Footsteps are by far the most commonly-played sound in the game

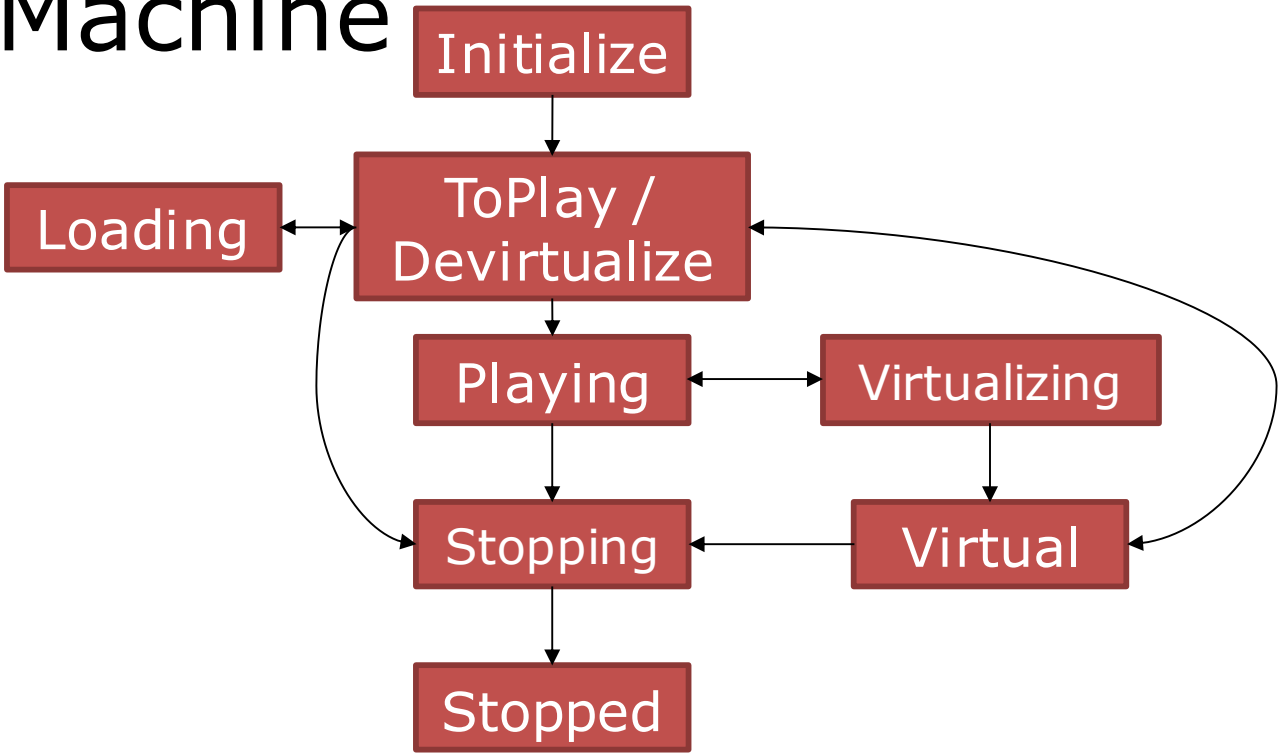
Footsteps

- You can go nuts
  - …but should you?

Summary

- If you remember nothing else from this talk, this is the most important slide…

State Machine

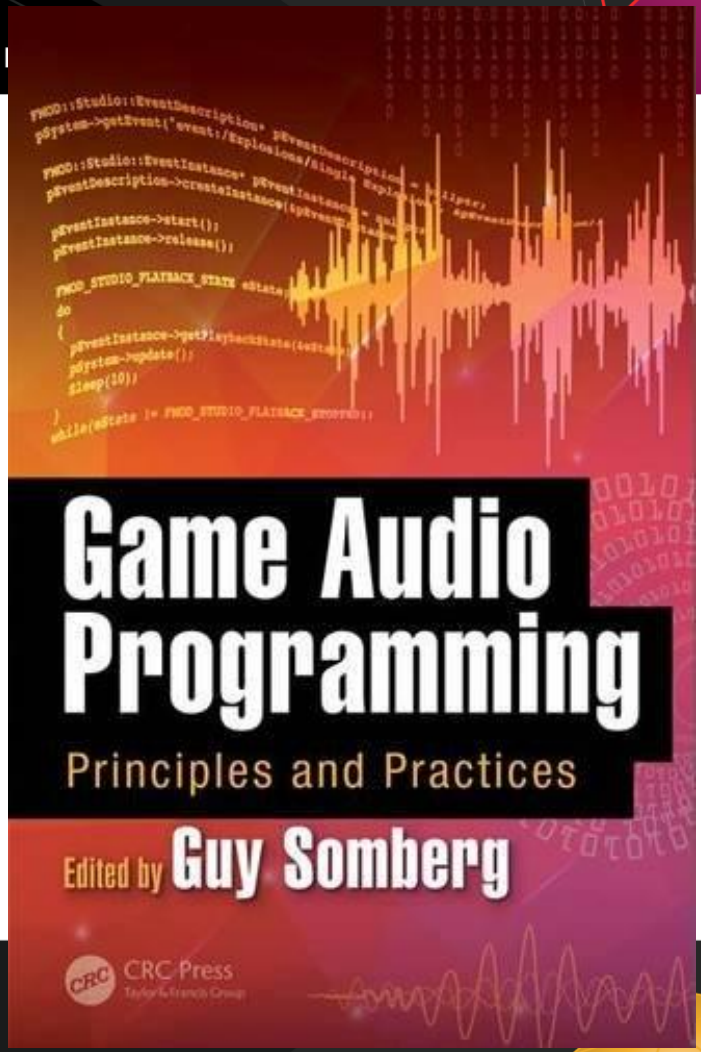


Summary

- Virtual sounds: A little bit of tech goes a long way
- Volume Sliders: Use the code that I showed
- Occlusion: Choose the method that works best for your game
- Footsteps: Never has so much effort been spent on a feature that so many people care so little about

Shameless Plug

CRC Press
Expo Booth #301



Questions?

- guy@gameaudioprogrammer.com

- Also, come to my Audio Programming Roundtables!
  Fundamentals ~~Wednesday 2:00-3:00~~
  Advanced Topics Thursday 4:00-5:00
  Free-For-All Friday 11:30-12:30