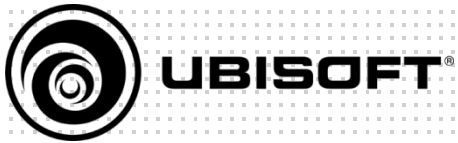
Real-time Sound propagation in video game

Jean-François Guay, Ubisoft Montreal
Jean-francois.guay@ubisoft.com



GAME DEVELOPERS CONFERENCE 2012
SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9

Realistic rendering can increase immersion of the player

But what about sound?

Agenda

- Physics of wave interaction
- What others have done
- Limitations of the game industry
- Our way of doing it
- Problems we faced
- Real examples

A bit of Physics

- Wave interaction effects
  - Reflection
  - Absorption
  - Diffraction
  - Refraction

A bit of Physics

- Audible sound is between 20Hz and 20 KHz
- Corresponds to a wave length between 17 m (low frequency) and 2 cm (high frequency)

Wave interaction

Reflection

Sound	Light
Reverb, echo	Specular reflection

Wave interaction

Absorption

Sound

Occlusion by a door or window

Light

Semi-transparent material

Wave interaction

Refraction

Sound

Rare: Sound that can be heard farther over a lake in the morning

Light

Broken straw in a glass full of water

Wave interaction

Diffraction

Sound

Hearing sound not in line of sight

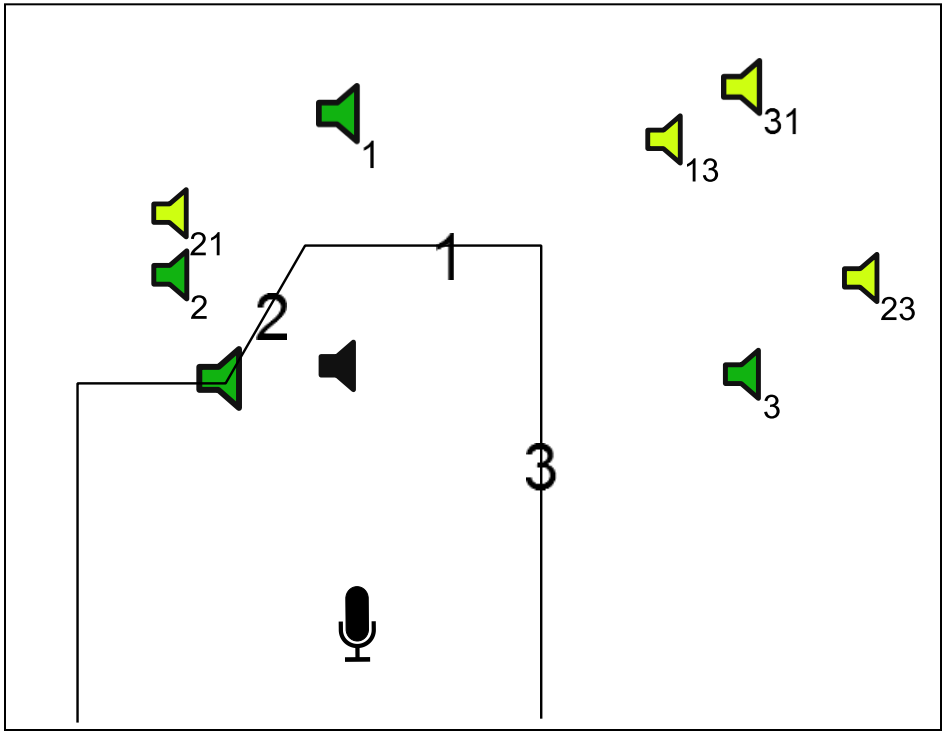
Light

Rare:

iridescent cloud, color on a CD

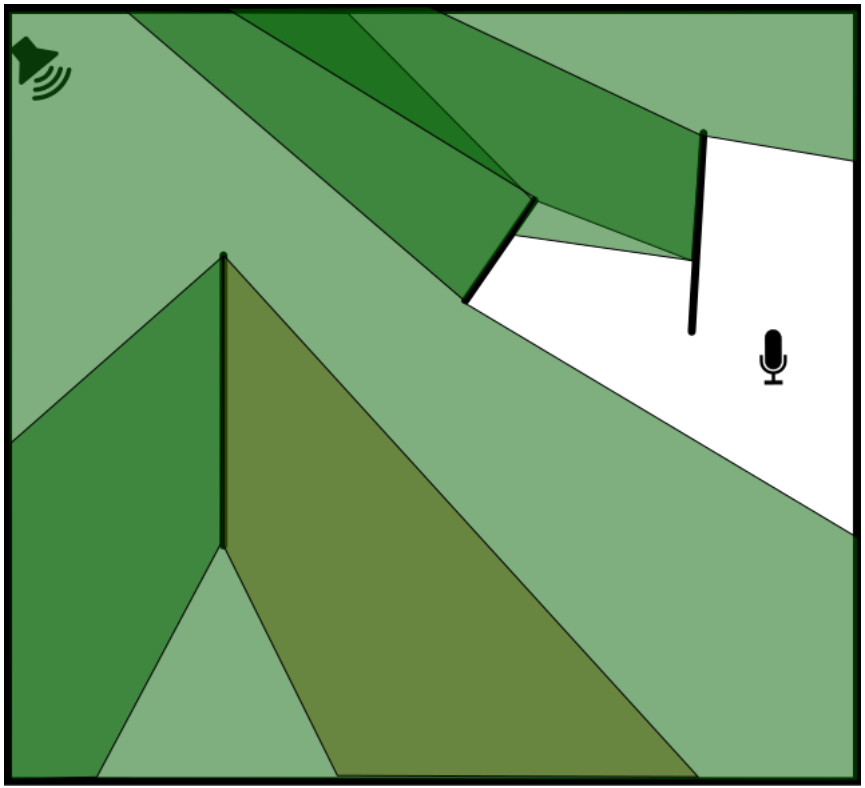
Different way to render virtual soundscape

Virtual source method



Different way to render virtual soundscape

Ray Tracing and
Beam Tracing



Limitations of game industry

- Game wants to run between 30 and 60 FPS
- Cannot "downgrade" the game for sound propagation
- Limited memory (2 to 50 MB for audio)
- Limited CPU (around 10% of the total CPU for audio)
- Up to 64 simultaneous sounds playing
  - At 60 FPS, that gives 250 μs per sound

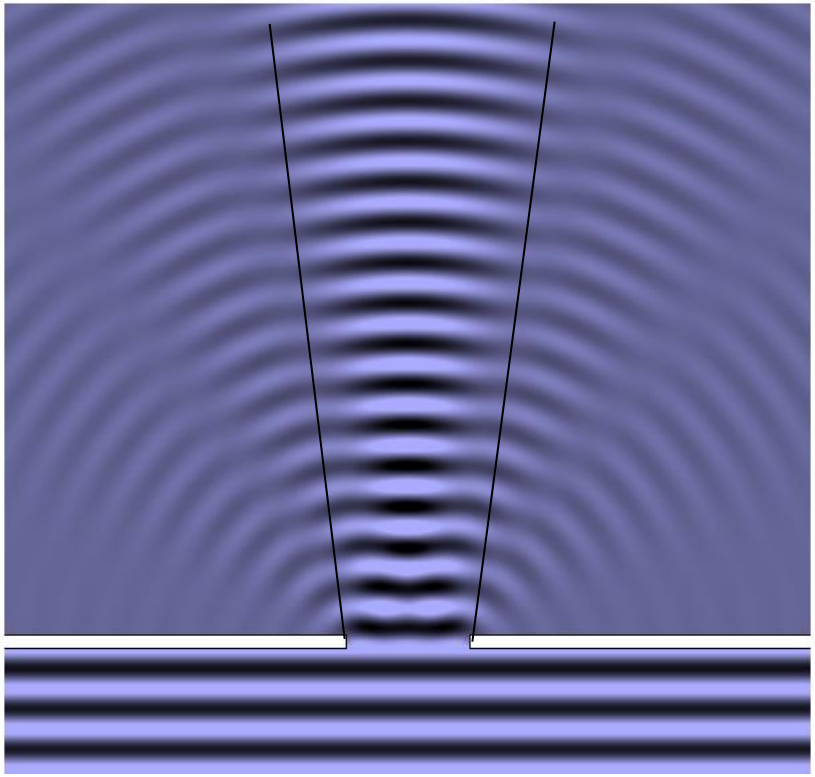
Goals we want to achieve

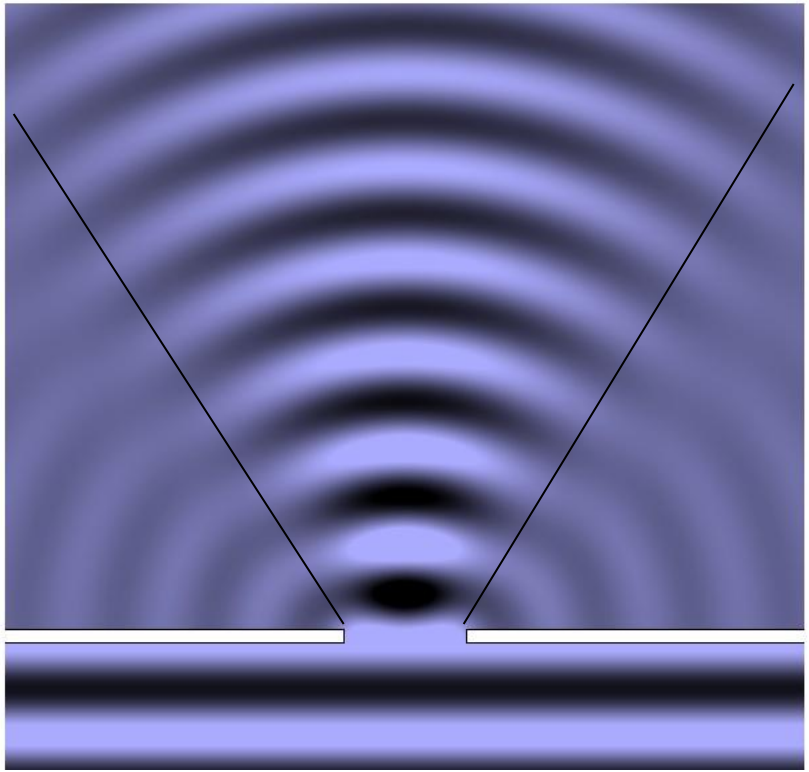
- Immersive result, not really interested in a physically exact result

Goals we want to achieve

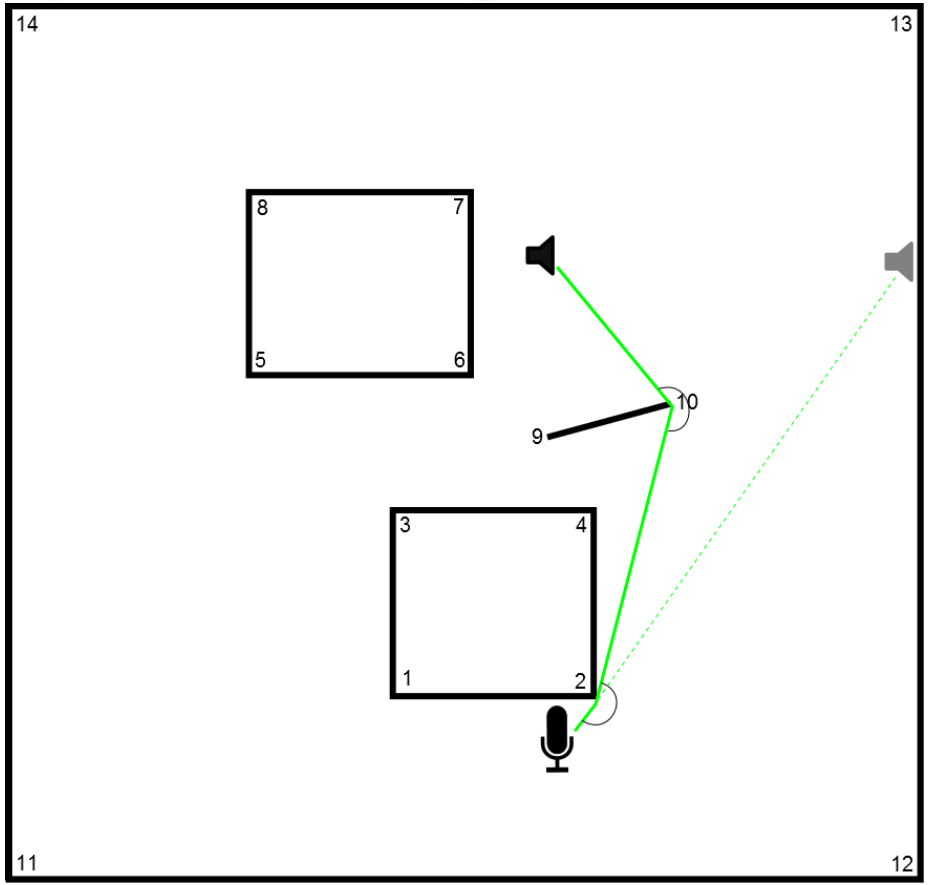
- Immersive result, not really interested in a physically exact result
- Current reverb algorithm already correctly simulates reflection of sound in the room
- Wanted to focus on diffraction, absorption and getting the general direction from where the sound is coming

Diffraction



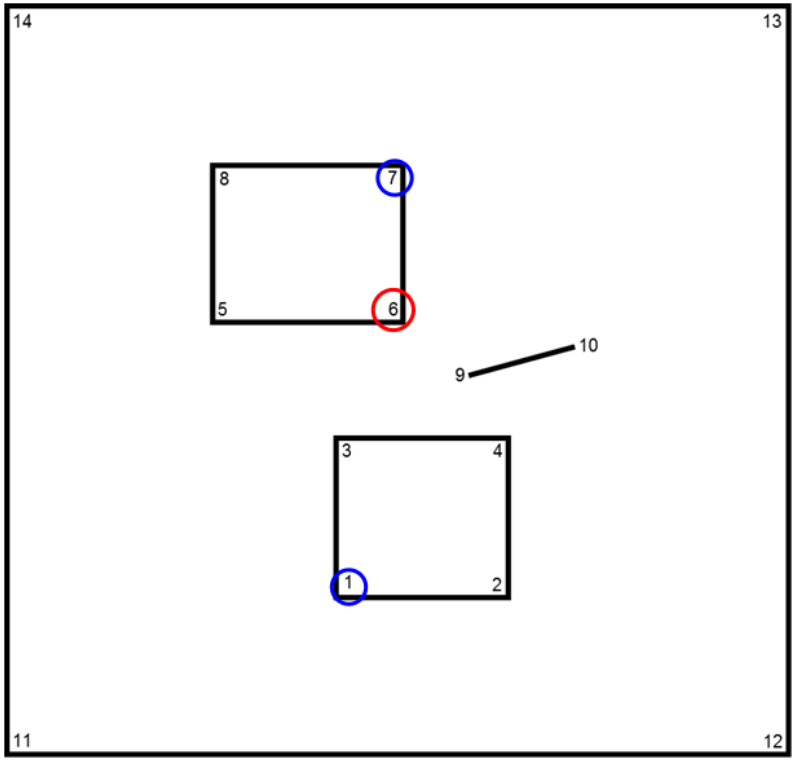


Our way of doing it



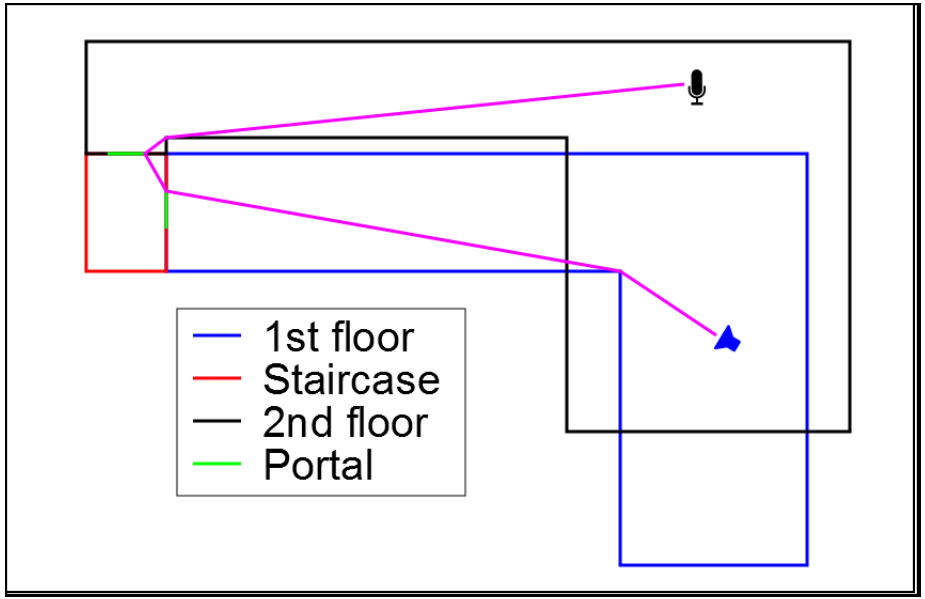
Our way of doing it

Edge dist	1	2	3	4	5	6	7
1	-- -	2 10	3 10	2 20	5 23	3 17	3 27
2		--	1 20	4 10	1 35	4 21	4 36
3			--	4 10	5 13	6 7	6 17
4				--	5 17	6 11	9 3
5					--	6 10	6 20
6						--	7 10
7							--



Optimization

- Zoning to remove irrelevant geometric data
- Computation done in 2D + height



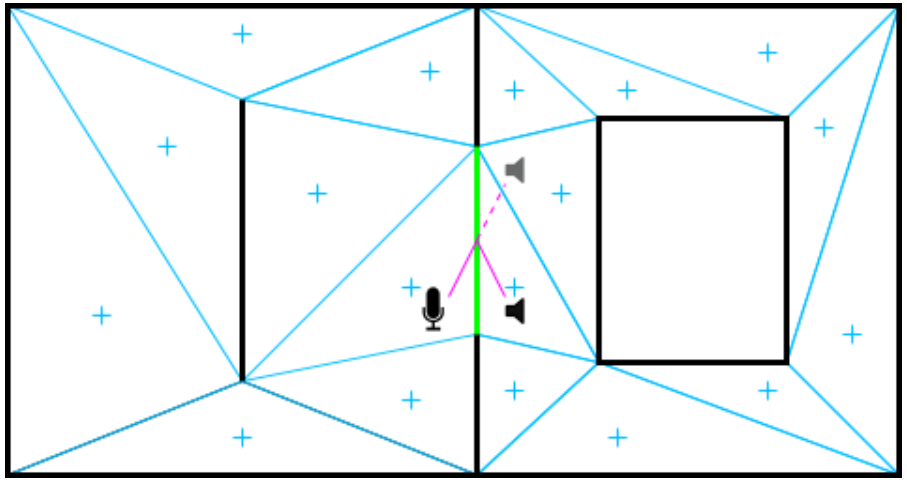
Optimization

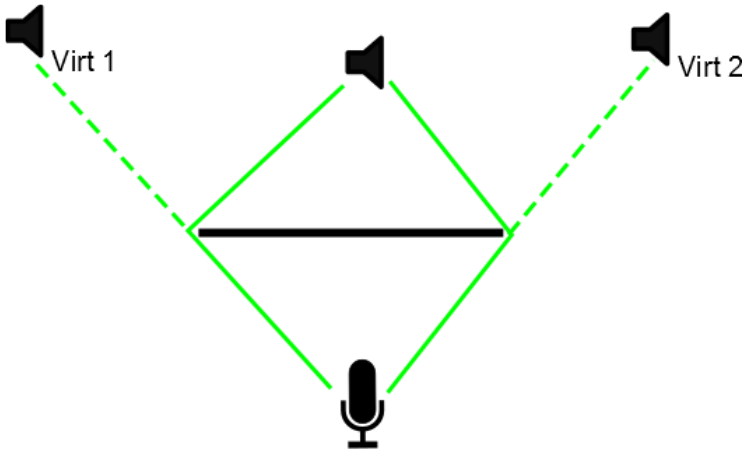
- Zoning to remove irrelevant geometric data
- Computation done in 2D + height
- Added a “microphone reception distance” that results in ignoring walls outside this range
- Frame rate of the algorithm is independent of the game frame rate
  - FPS between 5-10 is normally sufficient
  - We first compute objects that have moved the most

Problems we faced

Not the same as AI Path finding

Discontinuity





Problems we faced

Dynamic loading

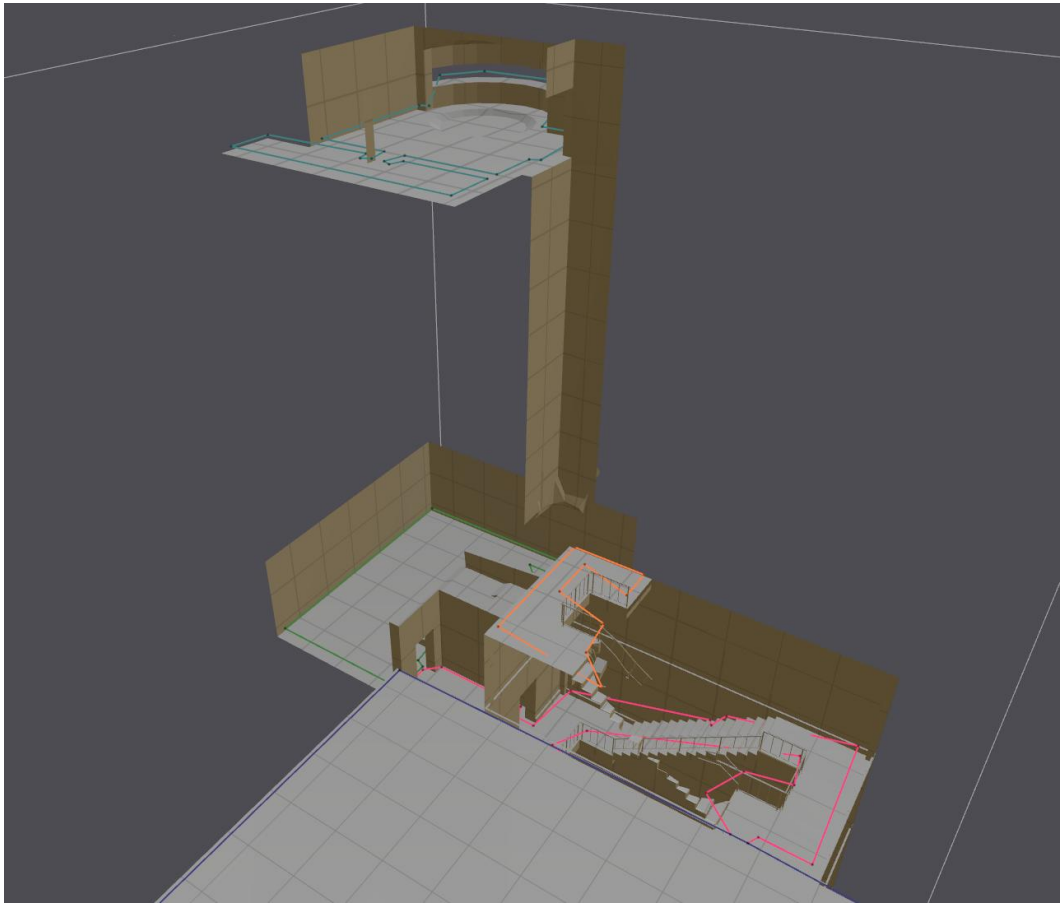
- Had to reorganize data structure for pre-computed shortest paths

Problems we faced

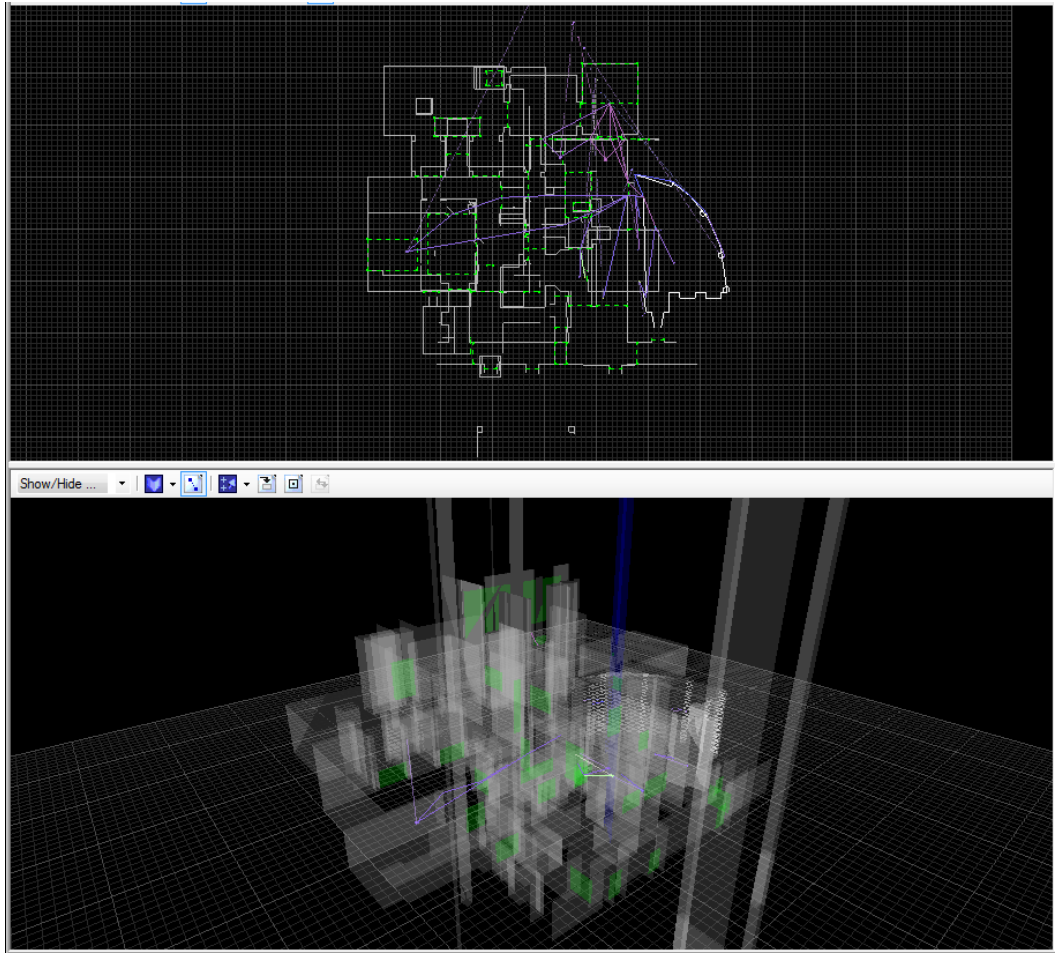
Data creation

- First version needed to flag walls individually
- Was hard to have a comprehensive, 100% automatic solution
- Ended up with 2 automatic generation algorithms
  - One optimized for complex interiors
  - One optimized for exteriors with small buildings

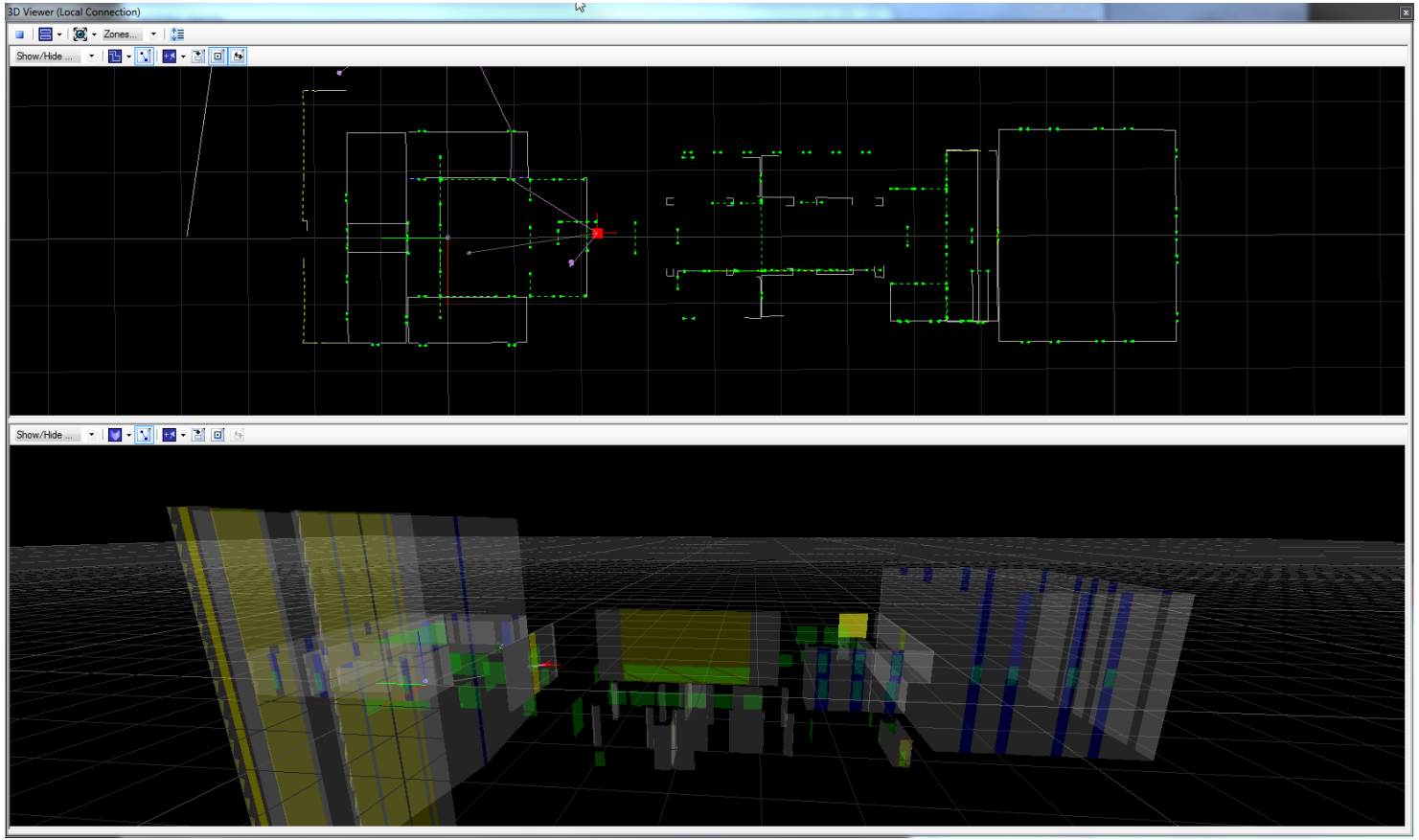
Data creation



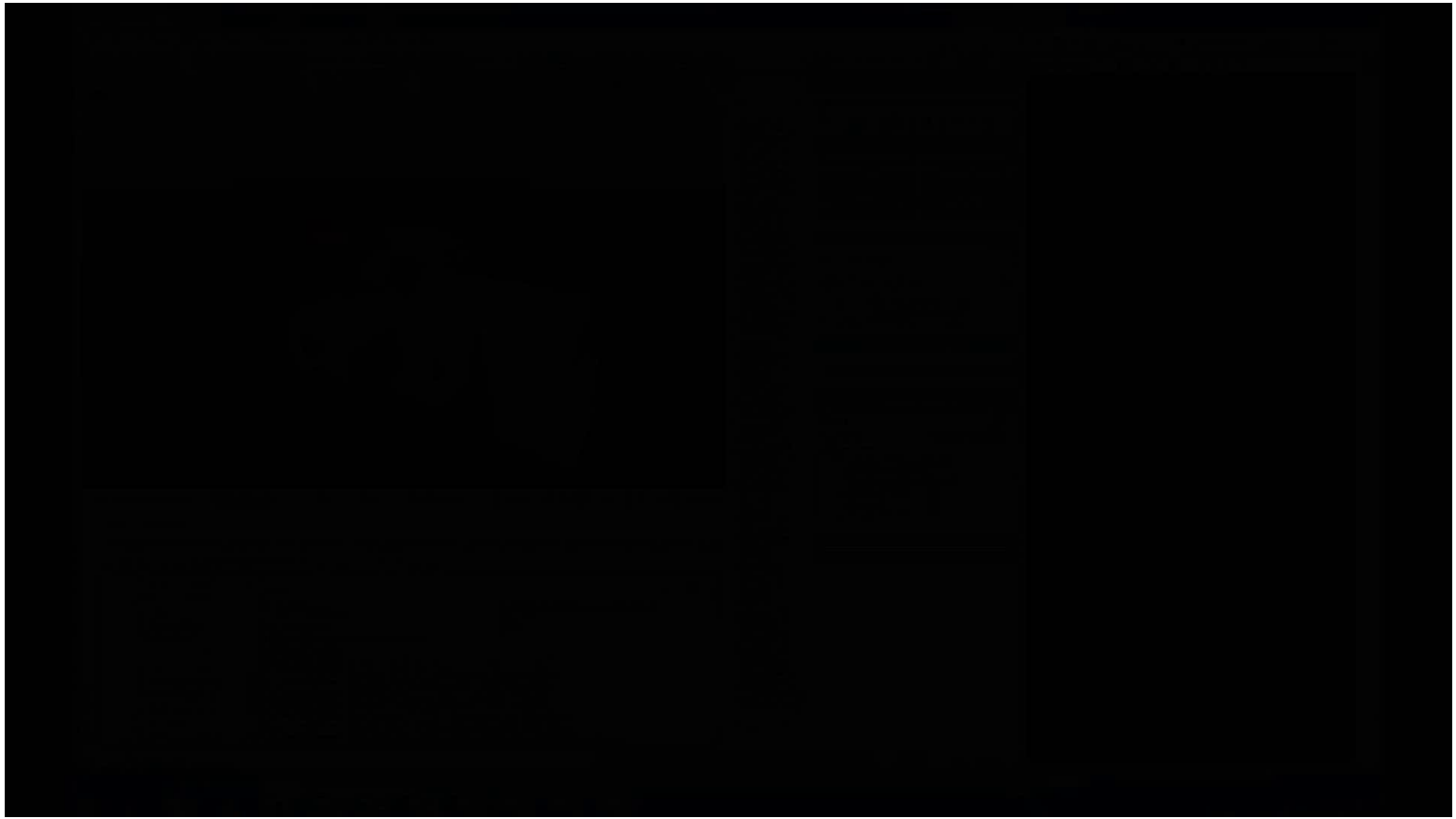
Data creation



Data creation

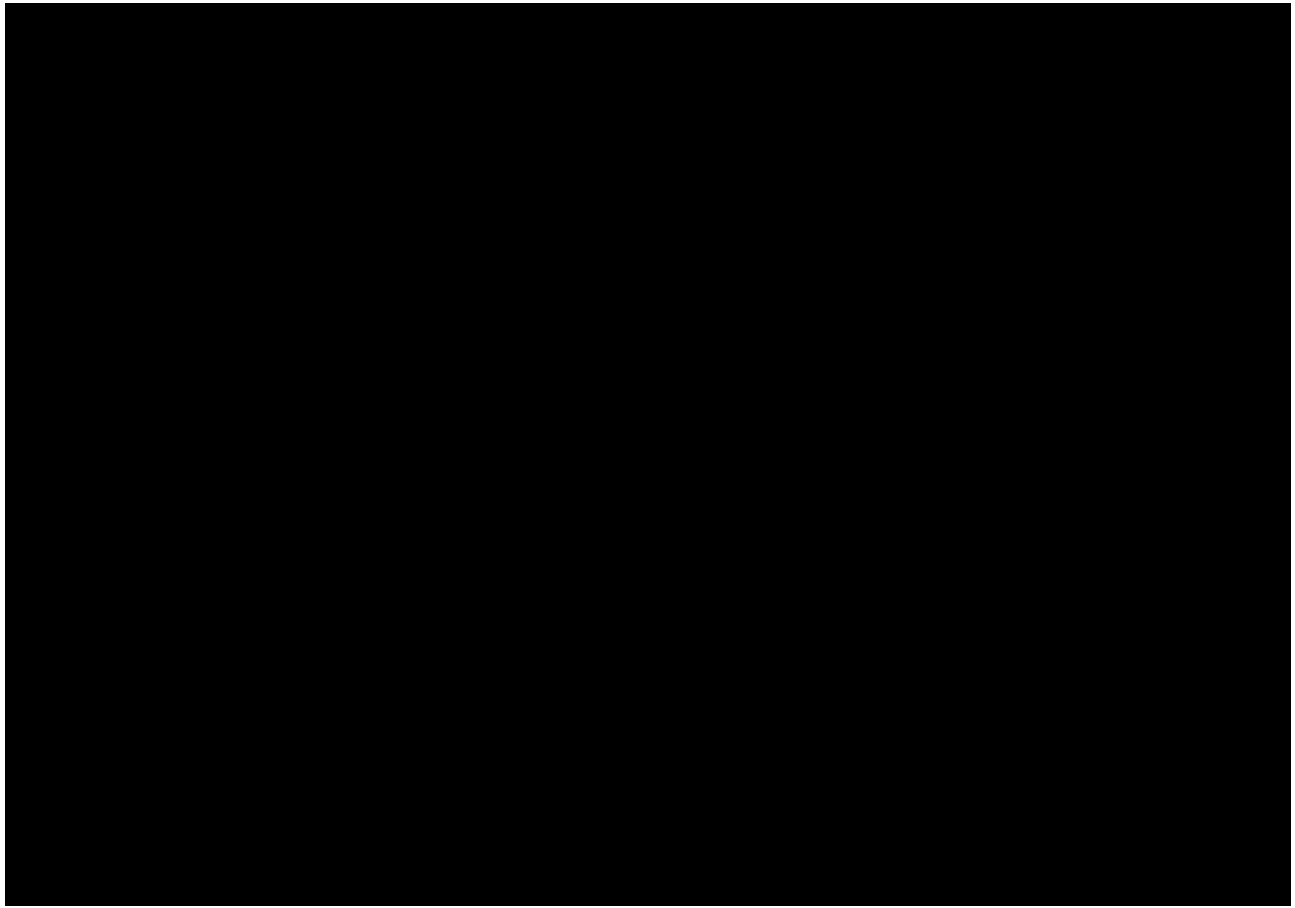


Data creation



Some real examples (diffraction)

Some real examples (absorption)



Splinter Cell Conviction

Sound Propagation integration

Future developments

- Dynamic/destructible environments
- Improve automatic generation

Conclusion

- Needs to be simple to use and fast
- Should not have discontinuity
- Audio designer needs to have control on how it will be rendered
- Propagation is a good way to increase player's immersion
- Even possible to do game play with propagation

engine room.ubi.com

powered by UBISOFT

YOUR ONE-STOP IN-DEPTH TECH BLOG

Thank you

Questions ?

Contact email:
Jean-francois.guay@ubisoft.com