CREATING IMMERSIVE AUDIO EFFECTS IN GAMES AND APPLICATIONS USING VRWORKS AUDIO

Ambrish Dantrey
Tony Scudiero

NVIDIA
AGENDA

VRWorks Audio
What’s New
C API
Unreal Engine 4 Plugin
Best Practices
Performance
Virtual Reality
Psychoacoustics
NVIDIA VRWORKS AUDIO
VRWORKS AUDIO
GPU-accelerated, Real time, Ray-traced Audio

- Real-time modeling of the following effects:
  - Sound propagation, direct and indirect paths
  - Occlusion for direct and indirect paths
  - Directionality/HRTF
  - Attenuation
  - Diffraction Approximation
  - Dynamic Scenes- Geometry / Material
  - Material reflection, absorption, & transmission
VRWORKS AUDIO SDK AND PLUGINS

GPU-accelerated, Real time, Ray-traced Audio

SDK

C API

Enables all application developers

Game Engine Plugin

Unreal Engine 4 Plugin

Enables UE4 game developers
WHAT’S NEW
VRWORKS AUDIO EVOLUTION

- **Inception**
  - Early-Access Builds
  - Private Demos
  - H1, 2016

- **SDK 1.0**
  - UE4.15.1
  - Pascal
  - Synchronous API
  - H1, 2017

- **SDK 1.1**
  - UE4.15.1
  - Pascal
  - Asynchronous API
  - H2, 2017

- **SDK 2.0 beta**
  - UE4.18
  - Volta
  - Performance++
  - Private Access
  - H1, 2018
VRWORKS AUDIO SDK 1.0

GPU-accelerated, Real time, Ray-traced Immersive Audio

- Adjustable quality and compute presets
- Dynamic geometry
- Dynamic materials
- Synchronous APIs
- Unreal Engine 4.15 Plugin
VRWORKS AUDIO SDK 2.0 BETA

GPU-accelerated, Real time, Ray-traced Immersive Audio

- Asynchronous APIs
- Increased number of VRAudio Sources
- Unreal Engine 4.18 Plugin
  - New UE4 audio path

*Currently Private Access Only*
C API
VRWORKS AUDIO PIPELINE

Application

Scene Information

Simulation

Filters

Filter Application

VRWorks Audio

‘Dry’ Audio

‘Wet’ Audio
Application Flow

Init
- Initialize library
- Create NvAR named context

Setup
- Choose compute presets (low compute/high compute)
- Provide context parameters (mesh, source, listener, effect strength presets etc.)

Filter Gen
- Generate convolution filters (one each for L, R ear)
- 1 filter per VRAudio source/listener pair
- Represents directionality, reverbs, transmission, diffraction, occlusion effects in one filter

Apply
- Apply filter to audio waveform for each VRAudio-enabled audio source

Cleanup
- Generic API cleanup
Application Flow – API Reference

Global Init:
   nvarInitialize();
   nvarCreate();

Global Setup:
   nvarCreateMaterials();
   nvarCreateMesh();

Game Loop:
   Setup:
   nvarCreateSource();  // Add sources (when applicable)
   nvarSetSource();     // Change source position and orientation
   nvarSetListener();   // Change listener position and orientation
   nvarTransformMesh(); // Move geometry within the scene (where applicable)

   Filter Gen:
   nvarTraceAudio();   // Generate filters

   Filter Apply:
   nvarApplySourceFilters();  // Apply effects to audio stream
   nvarApplySourceDirectPathFilter() // Apply only HRTF
   nvarApplySourceIndirectPathFilter() // Apply only reverb
   nvarSourceSubmitBuffers()    // Load dry audio
   nvarApplyIndirectPathFiltersToSubmittedBuffers()  // Get mixed wet audio

   Clean-up:
   nvarDestroySource();  // Terminate a source (when applicable)

Global Clean-up:
   nvarDestroyMesh();
   nvarDestroy();
   nvarFinalize();
GLOBAL INITIALIZATION
Initialization – Global

Library Initialization APIs

- `nvarInitialize()`
  
  Initializes the API. Call once before calling any other API.

- `nvarCreate()`
  
  Create named context. Set high or low compute.
GLOBAL SETUP
Setup – Materials

Material APIs

- **nvarCreateMaterial()**
  
  Create global set of materials used in scene/geometry

- **nvarSetMaterialReflection()**
  
  Set coefficients for audio reflection; 1.0 = fully reflective, 0.0 = fully absorptive

- **nvarSetMaterialTransmission()**
  
  Set coefficients for audio transmission; 1.0 = fully transmissive, 0.0 = fully reflective
Setup – Geometry

Geometry/Scene Definition APIs

• `nvar CreateMesh()`
  
  Defines a mesh in the scene/geometry. Can add mesh dynamically.

• `nvar CommitGeometry()`
  
  Pre-process geometry information for static scene/geometry. Optional for dynamic scenes/geometry.
Setup – Geometry
GAME LOOP
Source and Listener

**Listener**

- `nvarSetListenerLocation()`
- `nvarSetListenerOrientation()`

Set listener location and orientation; can be different for each trace

**Source**

- `nvarCreateSource()/nvarDestroySource()` – Create/destroy audio sources as needed
- `nvarSetSourceLocation()` – Set source location; can be different for each trace
- `nvarSetSourceOrientation()` – Set source orientation; can be different for each trace
Source and Listener

Geometry

Object1/Mesh1

Object2/Mesh2

Object4/Mesh4

SetListenerLocation()

Source1

CreateSource()
SetSourceLocation()
SetSourceOrientation()

Listener

Object3/Mesh3

SetListenerOrientation()

Source2

Curtain
Audio Effect Presets

- Low, Medium, High
- Reverb length & characteristics
- Per source, dynamic
- Subtle, Realistic, Accentuated
- Different smoothing coefficients, contribution factors
- All 3 presets available in low and high compute modes
- Specified in CreateSource()
Filter Generation

Pathtracing

- `nvarTraceAudio()`

  Generates 2 filter impulse responses: Left ear, right ear
  Called as frequently as possible
  Limited by GPU power and scene complexity

Asynchronous API: Event for synchronization
Application can optionally read filter
Filter Generation – Pathtracing
Using Generated Filter

Applying Effects - Apply filter to input PCM audio

• `nvarApplySourceFilters()` – NVIDIA-optimized filter convolution API
  • Apply effect L/R filters to source audio
  • Call this API from audio processing thread; e.g. XAudio2 callback
• Your choice of convolution function/API

Filter Impulse Response – Read filter’s impulse response directly

• `nvarGetSourceFilters()`
Using Generated Filter

**Applying Effects** - Apply filter to input PCM audio

- nvarApplySourceDirectPathFilter()
  - Apply HRTF effect to the mono audio to generate a L/R spatialized signal.
  - Call this API from audio processing thread.

- nvarApplySourceIndirectPathFilter()
  - Apply reverb effect to mono dry audio to generate a L/R wet audio.
  - Call this API from audio processing thread.
Using Generated Filter

Applying Effects - Apply filter to input PCM audio

- `nvarSourceSubmitBuffers()`
  - Submit mono dry audio to `nvar`

- `nvarApplyIndirectPathFiltersToSubmittedBuffers()`
  - Apply filter to mixed dry audio from multiple sources.
Dynamic Geometry

Change Geometry Elements “On-the-Fly”

- `nvarCreateMesh()`, `nvarDeleteMesh()`, `nvarTransformMesh`

  Dynamically add/delete/transform objects/meshes in the scene/geometry
Dynamic Geometry
CLEAN-UP
Cleanup

Clean-up

- `nvarDestroyMaterial()`
  Destroy material definition structures. Must match with `nvarCreateMaterial()`

- `nvarDestroyMesh()`
  Destroy geometry/scene meshes. Must match with `nvarCreateMesh()`

Debug

- `nvarExportOBJs()`
  Export scene/geometry meshes used by NVAR to .OBJ files
  e.g. “Is this wall visible to the VR audio library?”
Threading Model

Main/Game Thread

- NvAR Init
- NvAR Setup
- Game Loop
- NvAR Clean-up

Audio Processing Thread (Implicit in Sound Engine)

- Get Audio Data
- Apply Filter

Filter Generation Thread

- Update Source & Listener
- Filter Gen
UNREAL ENGINE PLUGIN
UE4 PLUGIN INTEGRATION FLOW

Global Setup
- Compute preset
  - High Compute
  - Low Compute
- Use Nvidia Physx Mesh Data
- Max VRAudio Sourced
- Reverb Length
- Units per meter
- Global Material

Material Setup
- Acoustic Mesh
- Disable NVAR
- VRAudio Material
  - Reflection Coefficient
  - Transition Coefficient
  - Absorption Coefficient

Audio Source Setup
- VRAudio Spatialization Source Settings
  - Direct path gain
  - Effect Preset
  - Low, Medium, High
- VRAudio Reverb Source Settings
  - Indirect path gain
Global Setup
Material Setup
Audio Source Setup
BEST PRACTICES
MAKING THE MOST OF IMMERSIVE AUDIO

• Chose correct performance preset
  • High Compute Vs Low Compute
• Chose correct effects preset
  • Low, Medium and High
• Use Units per meter to acoustically scale
• Use direct and indirect gain for fine-tuning
MAKING THE MOST OF IMMERSIVE AUDIO

- Spatialize sounds coming from the player in a single submix
  - Footsteps, speech, gunshots, breathing
- Prioritize speech and impact/explosion sounds for geometric effects
- Apply spatialization effects last in any DSP chain
OTHER ADVICE

As beautiful as it sounds (https://www.youtube.com/watch?v=e4dT8FJ2GE0&t=57), monster reverb isn’t always the most realistic.

Capture sounds as anechoically as possible

Balloon Pop - Stock Sound Effect

Balloon Pop - Recording in Anechoic Chamber
PERFORMANCE
TRACE TIME

High Compute

Low Compute

Measured on geometry with .5 million triangles
AUDIO & VIRTUAL REALITY
#actuallyLiterally

LITERALLY

VR IS ^GAME CHANGING

Making and experiencing VR is fundamentally different than traditional gaming.
PSYCHOPHYSICS

Sensation

A physical phenomenon

Psychophysics

Perception

purely mental phenomenon
GLOSSARY

**Stimulus** - Physical phenomenon - measure of environment using sensory organs

**Perception** - Mental phenomenon - (sub)conscious understanding/summary of stimulus

**Experience** - Mental phenomenon - Integral of perceptions over some time window $T$

$$\int_0^T \left[ \int_{pov} \text{VisionPercept}(p)dp \cdot \int_{-k}^K \text{HearingPercept}(k)dk \right] dt$$  \hspace{1cm} (1)

(1) This equation is made up and doesn’t mean anything
Warning: Nerdy Joke

Q’S
Warning: Nerdy Joke Continues

QUEUES

insert

remove
Warning: Compound Nerdy Joke: 2X nerd multiplier

QUEUES OF Q’S

insert

remove
**CUES**

Clues without an L

**Cue** - property, attribute, or information encoded in a stimulus that affects how a brain perceives it

*Classic Machine Learning*: Features Extraction

*Deep Learning*: First/convolution layer(s)

---

**Mentioned Deep Learning in VR Talk: +17 XP**

Next level in 18,210 XP
IMMERSION

Obligatory Video Game reference
Also, this game is awesome.
NEGATIVE CUES

Anchor user in reality, prevent immersion & suspension of disbelief

VR allows fewer negative cues than ever before

Must replace them with correct cues
VR & PSYCHOPHYSICS SUMMARY

Academic stuff is over now
There is no test

VR is sensory hijacking

Senses measure stimuli whose perception is affected by cues

Perceptions combine to experience

Negative cues/anchors diminish immersion in VR
NVIDIA VRWORKS AUDIO

Value Proposition

VRWorks Audio - A toolkit for creating audio stimuli in psychophysical agreement with a virtual environment

Geometric Acoustics - Physically-based acoustic simulation
NVIDIA VRWORKS AUDIO
NVIDIA VRWORKS AUDIO
NVIDIA VRWORKS AUDIO
RESOURCES

VRWorks Audio SDK: https://developer.nvidia.com/vrworks/vrworks-audio

UE4.15 Integration: https://github.com/NvPhysX/UnrealEngine/tree/VRWorks-Audio-4.15.1


Unreal Tournament Demo: https://www.youtube.com/watch?v=jVp88IjDX24

VRWorks Audio support: vraudio-support@nvidia.com