Why Sound?

- Emotional Impact
- Improved Presence
- Situational Awareness
- Sensory Substitution
- Better Graphics
- Product Recognition
3D Sound: Rendering Pipeline

• Emitter Model
  – How we represent sound sources

• Propagation
  – Modeling what happens to the sound once it leaves the emitter

• Localization
  – Creating the illusion of a positional source
Emitter Model

• Source Representation
  – How to represent the waveform produced by the source

• Source Intensity
  – Relative loudness of the source

• Radiation Pattern
Sample Playback

• Simplest approach to modeling an emitter is to use prerecorded sounds

• Use sound libraries or field recordings

• Problem:
  – Cannot easily modify sounds to match motion (ex. force of impact)
  – Sound files are typically large
Synthesis Techniques

• Use a procedural representation of sound
• Sound synthesis systems were developed primarily for musical applications
• Timbre Trees
  – one of the first attempts to parametrically synthesize sounds from motion parameters
• Sound signal is represented as functional composition
Timbre Trees

• Evaluating the tree at time $\tau$ produces a single sample

• Video

\[
(sine (+ freq (* 100 (sine (* 1.5 t)))))
\]
Physically-based Synthesis

• Idea is to generate sounds automatically from 3D models using dynamic simulation
• O’Brien, Cook, Essl – FEM data generated for deformable body simulator was used to calculate sound waves
• Doel, Kry, Pai – FoleyAutomatic: modal resynthesis based on contact data
Source Intensity

- Decibels (dB) \[ dB = 10 \log_{10} \left( \frac{I_2}{I_1} \right) \]
- Dimensionless, relative, logarithmic
- \[ I \propto A^2 \]
- dB pressure level \[ dB = 10 \log_{10} \left( \frac{p_2^2}{p_1^2} \right) \]
- \[ dB = 20 \log_{10} \left( \frac{p_2}{p_1} \right) \]
- dB SPL \[ dB = 20 \log_{10} \left( \frac{p}{20 \mu Pa} \right) \]
Source Intensity

- Radiation Pattern
  - Usually represented as a set of concentric cones
Spatialization

• Spatialization is the process of recreating auditory cues in order to create the illusion of a positional sound source

• In order to spatialize sounds we must:
  – Recreate distance cues
  – Recreate position cues
Distance Cues

• The intensity of a sound is the primary cue used to judge distance
  – Problem is that a listener’s familiarity with a sound influences this judgment

• Spectral composition of sound is also used to judge distance
  – High frequency components dissipate faster

• R/D ratio
Propagation Effects: Spreading Loss

- Sound traveling in free field conditions dissipates according to the inverse square law $1/r^2$
Propagation Effects: Spreading Loss

- We rarely hear point sources in free field conditions
- Surfaces near the source limit radiation pattern
- Reflected sounds reaching the listener greatly increases the energy reaching the listener
Propagation Effects: Spreading Loss

• Implementation of spreading loss
  – \( I \propto A^2 \)
  – Multiply waveform by \( D = \sqrt{\frac{1}{4\pi d^2}} = \frac{1}{3.55d} \)

• This does not take energy of reflected sound into account
Propagation Effects: Absorption

- Absorption occurs due to air particle friction
- Amount of energy lost is frequency dependent: higher frequencies result in higher friction
- Can use a low pass filter to simulate absorption
Propagation Effects: Refraction

• Atmospheric refraction can greatly effect the audibility of sounds in an outdoor environment
• Temperature inversion causes sound waves to bend back to earth
  – Sound velocity is greater in warmer air
• Wind speed gradients also cause to refract: Sound velocity is also effected by wind
Propagation Effects: Reverberation

• Similar to light, sound is a wave phenomenon that exhibits reflection, refraction, absorption and inverse square attenuation

• Unlike light, sound also can diffract around obstacles on a human scale and travel through nearly any barrier
Propagation Effects: Reverberation

- Sound energy reaches a listener via direct and reflected paths
- Order of reflection is the number of bounces before reaching the listener
Propagation Effects: Reverberation

- Room impulse response is a characteristic curve showing the reverberation characteristics of a room.
Propagation Effects: Reverberation

• Simulating sound propagation is a difficult problem
  – Main approach utilizes ray tracing from the source through the environment to find occlusions, first and second order reflections.
  – Further reflections are approximated by a reverberant tail
Position Cues

• The human auditory system localizes position of sound based on
  – Head Related Transfer Functions (HRTF)
    • Pinnea response
    • Shoulder echo
  – Interaural Time Difference (ITD)
  – Head shadowing or Interaural Intensity Difference (IID)
Localizing Sounds

• There are two predominant methods for spatializing sounds both are empirical methods:
  – Binaural techniques recreate HRTF, ITD and IID effects
  – Speaker panning techniques recreate the sound field by panning sounds among a set of speakers surrounding the listener
Binaural Techniques

• HRTFs can be measured directly by
  – Placing probe microphones in a listener’s ears
  – A pulse is played over set of speakers placed at positions surrounding the listener
  – The sound reaching the probe microphone inside the listener’s ear represents the effect of HRTFs for that source position
  – This can be encoded as a FIR filter
Binaural Techniques

- HRTF at 0°, 10°, 20° and 30° elevation
Binaural Techniques

• To place a sound in a location
  – For each ear
    • Find the 4 measured HRTF filters surrounding that location
    • Find the filter coefficients for the source location by interpolation the 4 surrounding filters’ coefficients
    • Apply the resultant filter to the source
Binaural Techniques

• ITD & IID
  – IID is normally encoded in the HRTF
  – To recreate ITD
    • Calculate the delay from source to each ear
    • Using a delay line apply the delay to left and right output channels

• When heard over headphones, the result is an impression of a positional source
Binaural Techniques

• Problems
  – HRTFs often result in internalization
    • Sounds appear to be inside the listener’s head
  – When sounds are externalized, HRTFs still do not recreate the impression of a distance source
  – Front-back reversals are also common where a sound in front of the listener is perceived to be in the back
  – These problems can be improved by measuring customized HRTFs for each listener
Panning Techniques

• Instead of recreating HRTF, ITD and IID effects, we can recreate the sound field directly by surrounding the listener with a set of speakers

• In order to spatialize a sound, it is panned between the speakers surrounding that position

• Stereo is a 1D speaker panning technique
Panning Techniques

• We can extend stereo to three dimensions by using 8 speakers and panning the source between those speakers

• Two panning algorithms
  – Constant intensity
    • Maintains a constant intensity of sound across the pan
  – Vector Based Amplitude panning
    • Uses any number of speakers panning between speaker triplets
Panning Techniques

• Problems
  – Panning techniques cannot generally place sounds inside the speaker enclosure (listening space)
  – Technique gives only a weak impression of a sound’s location
  – Speaker panning doesn’t reproduce correct elevation cues
Panning Techniques

• Actual source

• Panned source
Binaural Recording

- Record sound using 2 microphones implanted in a dummy head
- Recreates binaural effects when heard over headphones
Sound Hardware

• PC Sound Cards
  – ISA with FM synthesis
  – ISA with Wavetable synthesis
  – PCI with Wavetable synthesis
    • Support for DLS standard
  – Current cards provide hardware acceleration of 4 speaker panning, HRTF, Dolby 5.1 decoding
Sound Hardware

• Pro Audio Cards
  – Early systems used proprietary interfaces to output audio to an external A/D box
  – ADAT Lightpipe technology provided a standard to link pc and external A/D box
  – Current generation uses Firewire to shuttle digital audio back and forth
Specifications

• Sampling Rate
  – 22050, 44100, 96K
  – Nyquist theorem

• Sample Width
  – 8, 16, 20, 24
  – Quantization Error
  – S/N = 6n
API: General

- DirectSound & OpenAL
- Use the audio buffer as a first class modeling entity
- Support a single listener
- Make use of hardware acceleration
- Make use of EAX extensions
API: OpenAL

• Not an “open” version of SGI’s AL library
• Provides a GL like syntax for sound
• Main advantage: cross platform support
• OpenAL Specifies API but not 3D audio implementation
API: OpenAL

• Create Context
  – Device=alcOpenDevice((ALubyte*)"DirectSound3D");
  – Context=alcCreateContext(Device,NULL);
  – alcMakeContextCurrent(Context);

• Core OpenAL API operates under assumed context

• Audio library context provides OS bindings
  – ALC is portable across platforms
API: OpenAL

- **Only one listener: configure**
  - `alListenerfv`
    - Position, velocity, orientation

- **Create Buffers and fill them**
  - `alGenBuffers(NUM_BUFFERS, g_Buffers);`
  - `alutLoadWAVFile("footsteps.wav", &format, &data, &size, &freq, &loop);`
  - `alBufferData(g_Buffers[0], format, data, size, freq);`
  - `alutUnloadWAV(format, data, size, freq);`
API: OpenAL

• Create and configure sources
  – alGenSources(1,source);
  – alSourcef(source[0],AL_PITCH,1.0f)
    • Pitch, Gain, Position, Velocity, Looping

• Attach source to buffer
  – alSourcei(source[0], AL_BUFFER, g_Buffers[0]);

• Control play state
  – alSourcePlay(source[0]);
  – alSourceStop(source[0]);
API: DirectSound3D

• Microsoft’s DirectX Sound component
  – Uses capabilities found in sound cards to spatialize sounds
  – Uses software implementations when capabilities are not present in hardware

• In DirectX 8 3D sound through DirectSound or DirectMusic
API: DirectSound3D

• DirectSound Buffers
  – Hold waveform data
  – Application must provide waveform data for buffers
  – Buffers are manipulated through their interface
  – Have three interfaces
    • Standard buffer
    • 3D buffer
    • Property sets make DirectSound extensible
API: DirectSound3D

• Primary Buffer
  – One instance
  – Effectively the listener
  – All sources are mixed into primary buffer before sending to output device

• Secondary Buffers
  – One per sound source
  – Application fills with sound data
API: DirectSound3D

• Helper utils in dsutil encapsulate much of the buffer creation work

• Create and configure IDirectSound object
  
  ```cpp
  g_pSoundManager = new CSoundManager();
  ```

• Initialize
  
  ```cpp
  hr = g_pSoundManager->Initialize( hDlg, DSSCL_PRIORITY, 2, 22050, 16 );
  ```
API: DirectSound3D

• Get a pointer to the listener
  
  hr |= g_pSoundManager->Get3DLlistenerInterface( &g_pDSLlistener );

• Open a wave file and load it into buffer
  
  hr = g_pSoundManager->Create( &g_pSound, strFileName,
       DSBCAPS_CTRL3D, DS3DALG_HRTF_FULL );

• Control Sound

  g_pSound->Play( 0, DSBPLAY_LOOPING )

  g_pSound->Stop();

  g_pSound->Reset();

GWU
API: DirectSound3D

• Move Source
  
  memcpy( &g_dsBufferParams.vPosition, pvPosition, sizeof(D3DVECTOR) );
  
  memcpy( &g_dsBufferParams.vVelocity, pvVelocity, sizeof(D3DVECTOR) );
  
  g_pDS3DBuffer->SetAllParameters( &g_dsBufferParams, DS3D_IMMEDIATE );

• Or Listener
  
  g_pDSListener->SetAllParameters( &g_dsBufferParams, DS3D_IMMEDIATE );
API: EAX

- DirectSound & OpenAL only provide a distance model
  - Spreading loss
  - Absorption
- They do not model reverberation or sound occlusion and obstruction
API: EAX

• Developed by Creative, provides extensions to both APIs to model
  – Reverberation
  – Early reflection
  – Occlusion – source in a different room
  – Obstruction – object obstructing direct path between source and listener
API: EAX

- EAX extends DirectSound through property sets
  - Obtain an EAX interface for each secondary buffer and use it to control propagation model
  - Obtain EAX interface for the primary buffer (must do it through one of the secondary buffers) to control reverberation model
API: EAX

• EAX extends OpenAL through API extensions
  – Obtain addresses for extension functions: EAXSet and EAXGet
  – Set both listener and source EAX properties

• More Info:
Assignment

• Using either DirectSound or OpenAL, create a virtual sonic environment with at least one source and one listener

• Demonstrate:
  – Source control: start, stop
  – Source/listener motion: translation, rotation
  – Source occlusion/obstruction and room reverberation