Master IGMMV
Synthèse d’images et de sons

George Drettakis

Nicolas Tsingos

Programmation audio

Hardware Platforms

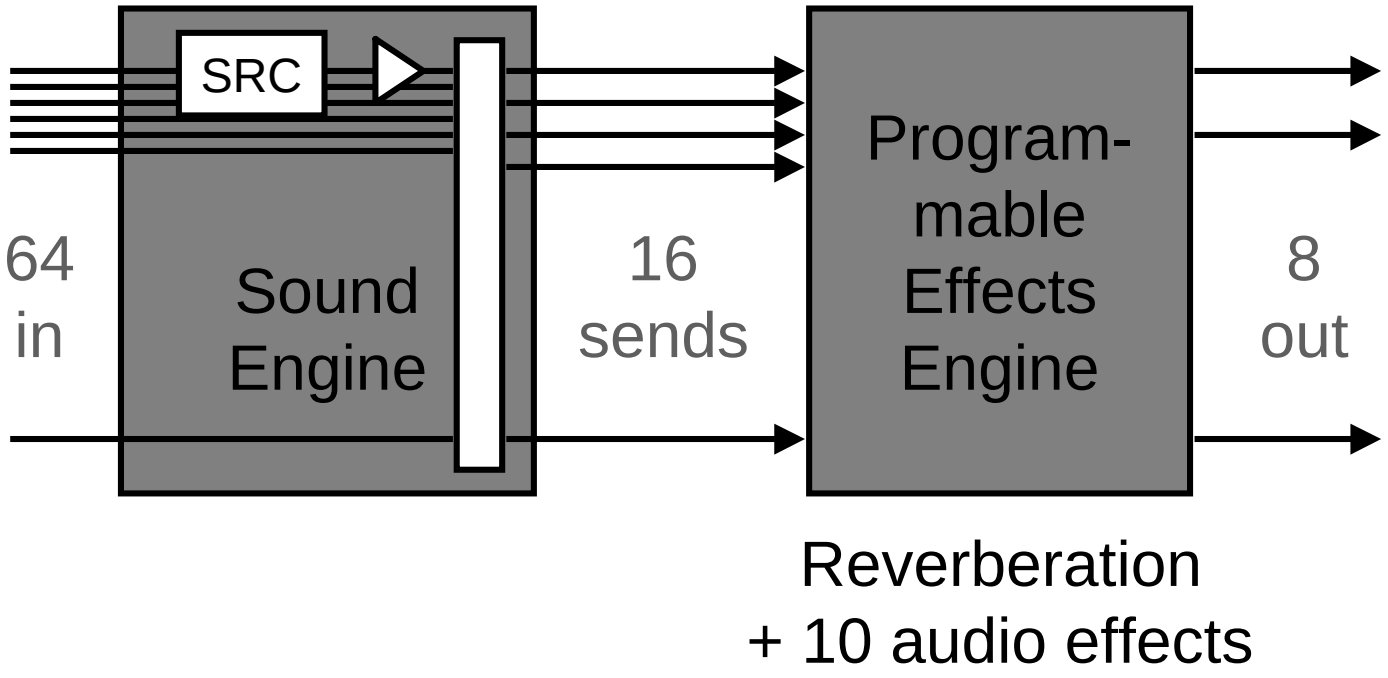
- Specialized DSP engines
  - Lake (low-latency convolution - long FIR filters)
  - Tucker Davis (3D audio support)
- Consumer-level PC sound cards & accelerators
  - DSound (+ EAX) widely supported on PC
  - Various levels of HW support:
    - I/O only
    - I/O + 3D positional
    - I/O + 3D positional + reverberation (& effects)

DSP / Rendering Technology

- Sample rate conversion - audio interpolation
  Variable delays, Doppler. Wavetable synthesis.
- 3D positional panning + mixing
  32-64 sources. Multiple configurations.
- Environmental reverberation
  “Glitchless” parameter updates.
- Musical / sound design effects
  2-D Premixing. Chorus, flanger, compressor...

DSP / Rendering Technology

- Example audio accelerator **(EMU10K1 chip)**



Interactive 3D Audio APIs & Standards

- General requirements
  - Positional representation independent from spatial reproduction technique and playback setup
    - Absolute or head-relative 3D coordinates
    - Cartesian or polar
  - Program-driven vs. description-driven applications
  - High-level vs. low-level scene description models

Direct Sound

- Partie audio de Direct X
  - gère les entrées-sorties audio temps-réel
  - support hardware par les cartes audio
- DirectSound object
  - configuration audio
- DirectSound primary buffer
- DirectSound secondary buffers
  - mixés dans le primary buffer

Direct Sound Object

- Device
  - carte son
- Configuration d ’écoute
  - casque
  - enceintes stéréo
  - multi-canal

DirectSound Object

```
HRESULT WINAPI
  DirectSoundCreate8(
  LPCGUID  lpcGuidDevice,
  LPDIRECTSOUND8 * ppDS8,
  LPUNKNOWN  pUnkOuter
);
```

Direct Sound Primary buffer

- Format de données audio
  - quantification
  - format (entiers, flottants)
  - fréquence d ’échantillonnage
  - nombre de canaux
  - compression
- Mode exclusif ou partagé
- Pas de contrôle sur le primary buffer

Direct Sound Secondary Buffers

- Tableaux circulaires de données échantillonnées
  - lus à la fréquence d ’échantillonnage audio
  - il faut écrire assez vite pour ne pas perdre des données
  - mixés dans le primary buffer
- Paramètres de contrôle
  - volume
  - position (buffers 3D)
  - pitch

Direct Sound Buffers : creation

```
typedef struct {
  WORD  wFormatTag;
  WORD  nChannels;
  DWORD nSamplesPerSec;
  DWORD nAvgBytesPerSec;
  WORD  nBlockAlign;
  WORD  wBitsPerSample;
  WORD  cbSize;
} WAVEFORMATEX;
```

```
HRESULT CreateSoundBuffer(
  LPCDSBUFFERDESC pcDSBufferDesc,
  LPDIRECTSOUNDBUFFER * ppDSBuffer,
  LPUNKNOWN pUnkOuter
);
```

```
typedef struct {
DWORD dwSize;
DWORD dwFlags;
DWORD dwBufferBytes;
DWORD dwReserved;
LPWAVEFORMATEX lpwfxFormat;
GUID guid3DAlgorithm; } DSBUFFERDESC ;
```

DSBCAPS_PRIMARYBUFFER

DSBCAPS_CTRL3D
DSBCAPS_CTRLPOSITIONNOTIFY
DSBCAPS_LOCHARDWARE

Direct Sound Buffers : playback

```
unsigned long datalen,datalen2;
LPVOID data,data2;

secondary_buffer->Lock(0,file.size,&data,&datalen,
&data2,&datalen2,0);
memcpy(data,file.data,file.size);
secondary_buffer->Unlock(data,datalen,data2,datalen2);

primary_buffer->Play(0,0,DSBPLAY_LOOPING);
secondary_buffer->Play(0,0,DSBPLAY_LOOPING);
```

Direct Sound Buffers : streaming

- Système de notification

```
result = buffer->QueryInterface(IID_IDirectSoundNotify8,
                                (void **)&DsNotify);

DsNotify->SetNotificationPositions(…)
```

- Attente passive : lock système
  - attend une position et débloque le sémaphore
  - WaitForMultipleObjects(…)
- puis lock => écriture => unlock

Direct Sound Secondary Buffer : attributes

```
HRESULT SetPosition(
  D3DVALUE x,
  D3DVALUE y,
  D3DVALUE z,
  DWORD dwApply
);
```

```
HRESULT SetVelocity(
  D3DVALUE x,
  D3DVALUE y,
  D3DVALUE z,
  DWORD dwApply
);
```

EAX : Environmental Audio Extensions

- Extension de Dsound/openAL par Creative Labs
  - le leader du marché de la carte son pour gamers
  - supporté en hardware par les SoundBlaster
  - l'interface d'EAX (2.0) est standardisé *IASIG*
- Contrôle de l'effet de salle (réverbération)
- Contrôle des occlusions/obstructions
- Effets de type studio dans EAX 4.0 (vocoder, chorus, …)

EAX

- SECONDARYBUFFER:: QueryInterface(IID_IKsPropertySet,(void **)&eaxListener);

- eaxListener->Set(DSPROPSETID_EAX_ListenerProperties, DSPROPERTY_EAXLISTENER_ROOM ,NULL, 0, &val, sizeof(LONG));

- LONG l = 200*log10(level);
- eax->Set(DSPROPSETID_EAX_BufferProperties, DSPROPERTY_EAXBUFFER_ROOM , NULL, 0, &l, sizeof(LONG));

OpenAL

- Très similaire à DirectSound
- Différences majeures
  - « multi-plateforme »
  - séparation des buffers et des sources
    (on le retrouve dans EAX v.4.0 sous Direct Sound)
  - une syntaxe plus lisible à la openGL
    - alSource(…)
    - alcOpenDevice(…)
    - ALfloat listenerPos[]={0.0,0.0,0.0};
      alListenerfv(AL_POSITION,listenerPos);

Scene Description & Rendering Models

- DS3D/OpenAL

Listener position, orientation, velocity

Source position, orientation, velocity

Source directivity

- Outside Volume
- Sound cones

Distance effects

- Roll-off factor
- Min, max distance

Scene Description & Rendering Models

- DS3D/OpenAL

Listener position,
orientation,
velocity

Source position,
orientation,
velocity

Source directivity

Outside Volume
Sound cones

Distance effects

Roll-off factor
Min, max distance

- EAX

Environmental
reverberation,
reflections

Obstruction, occlusion,
exclusion

Source directivity

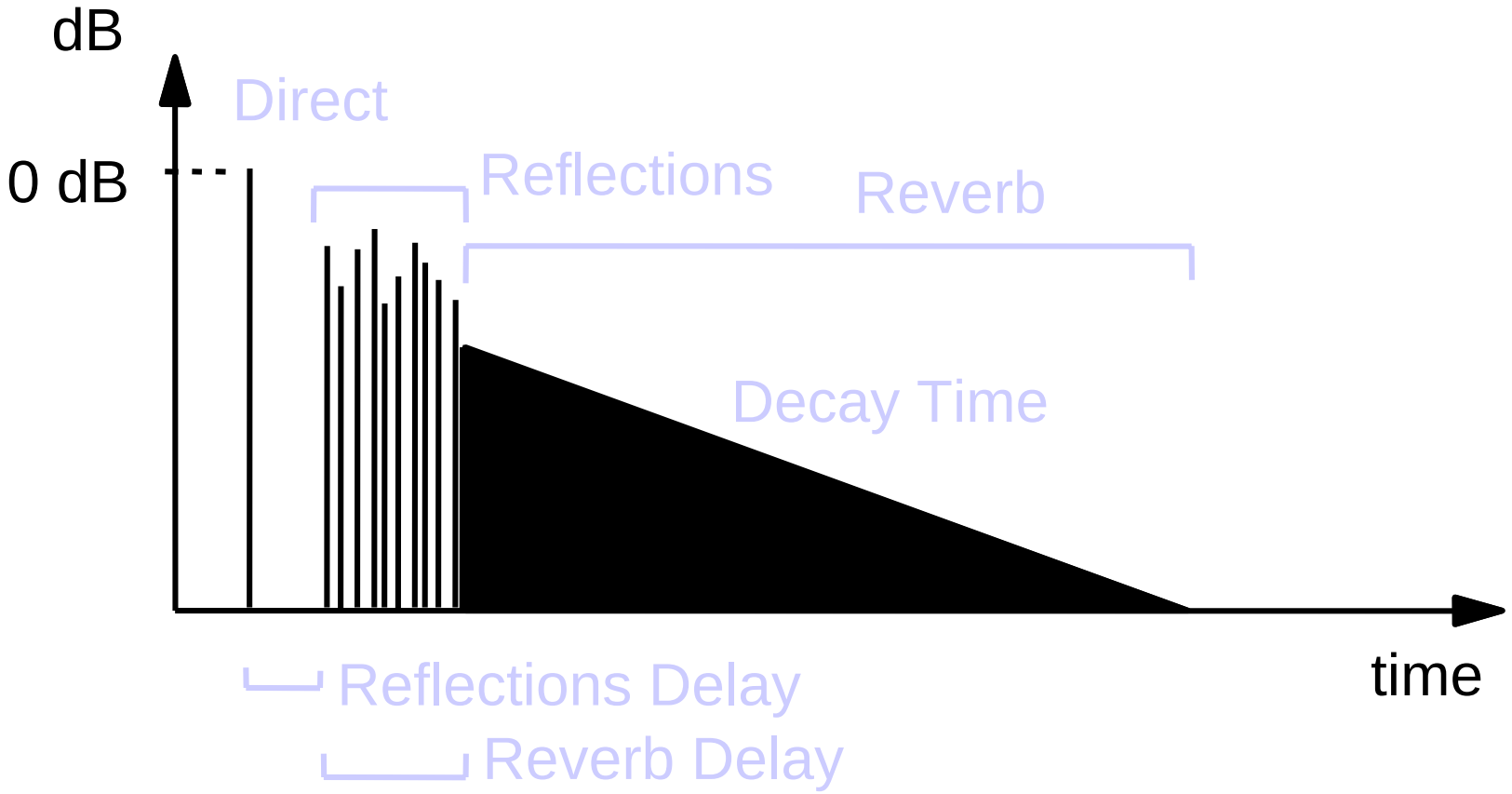
Outside Volume HF

Distance effects

Air absorption
Reverberation,
reflections

Scene Description & Rendering Models

- EAX/I3DL2 reverberation model



Scene Description & Rendering Models

- EAX: low-level model + statistical extensions

  Directional reflection cluster
  + directional reverberation tail

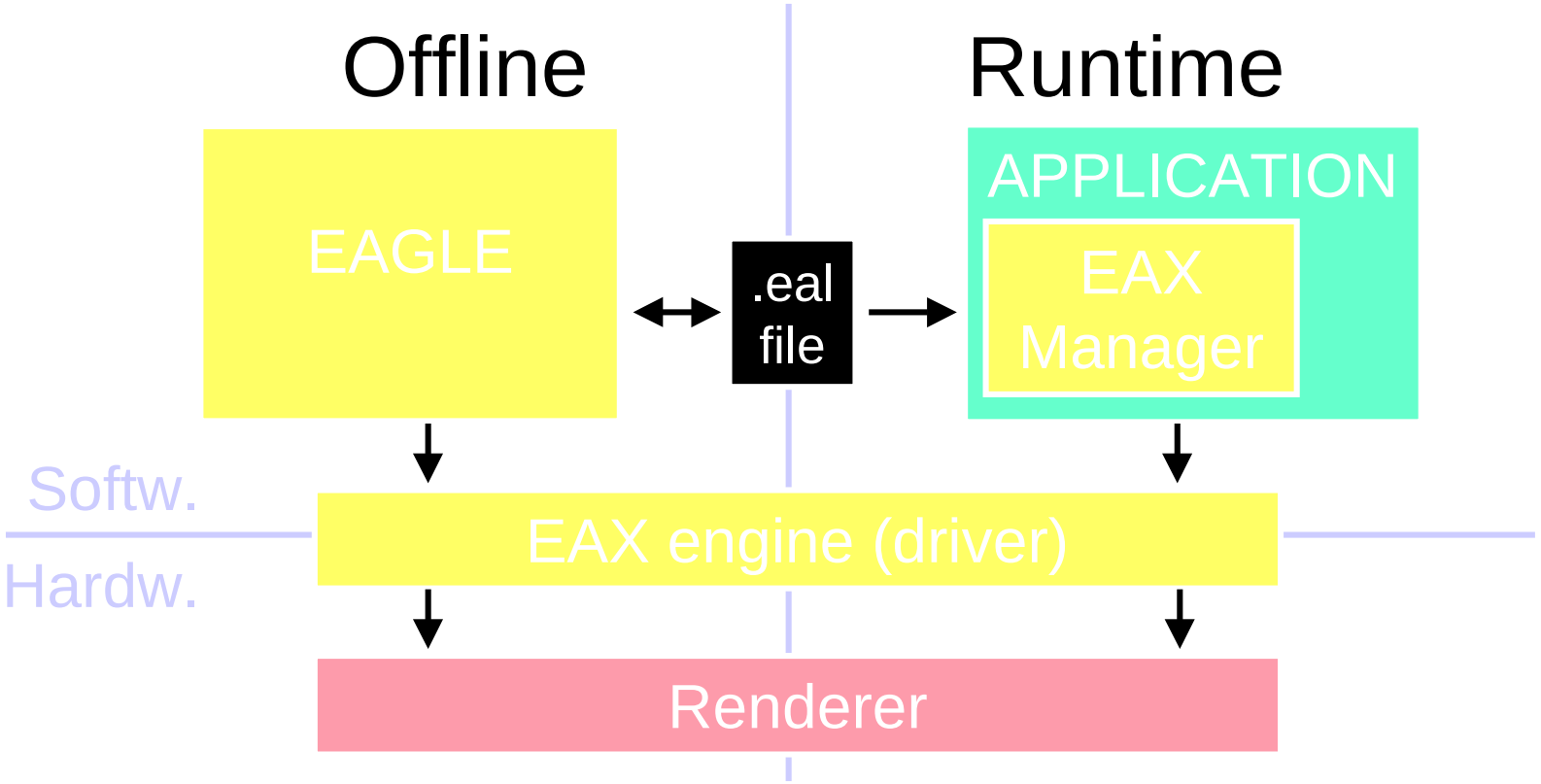
  Optional high-level statistical extensions

  distance an directivity models
  muffling effects
  room size control

Modeling/Authoring 3D Soundscapes

- Issues
  - Mapping world geometry to rendering parameters
  - Overcoming the constraints of physics
  - Collaboration sound designer / programmer
- An example: EAGLE and EAXManager API

EAGLE Editor & EAXManager API



EAGLE Editor

- For each source
  - Tune directivity and filtering parameters
  - Tune distance and Doppler parameters
- For each “room”
  - Design/tune environment and material presets
- Import geometry
  - Seal rooms
  - Tag rooms with environment/material presets
  - Define diffracting objects, tune diffraction model

EAXManager API

- Interrogation engine

  Queried by application at runtime

  Returns, according to source & listener positions:

  values of Environment properties

  values of Occlusion and Obstruction properties,
  position of virtual source (diffraction), etc…

- The application retains control of EAX

EAXManager API

- EAXManager audio engine
  - BSP technology
    - to locate listener and sound sources
  - Occlusion matrix
    - to look up occlusion properties for sounds
  - Axis aligned bounding boxes
    - to cull out objects that are not in the direct path
  - Ray tracing
    - to calculate obstruction