

Reprodução de Áudio

Programação de Jogos

Judson Santos Santiago

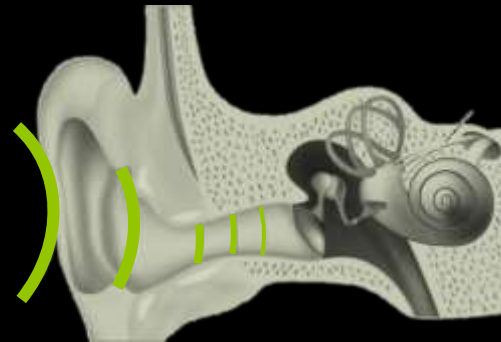
Introdução

► O que é áudio?

- Áudio é o som reproduzido por meios eletrônicos
 - Um som é produzido pela **variação de pressão** em um meio (como o ar)
 - Eles são criados pela **vibração do ar** ao redor de um objeto que vibra



Autofalante provoca
uma vibração no ar

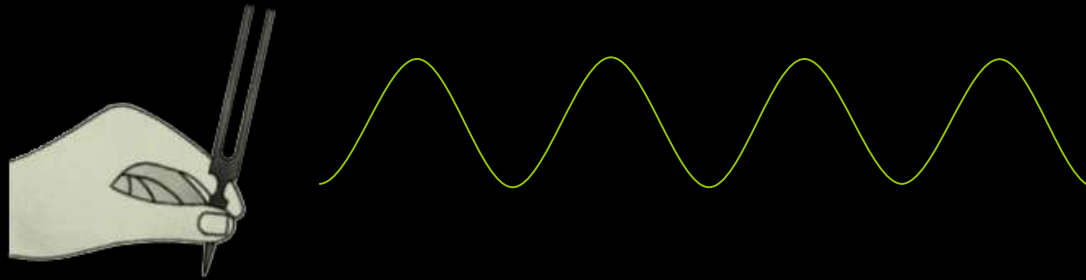


Membrana auricular

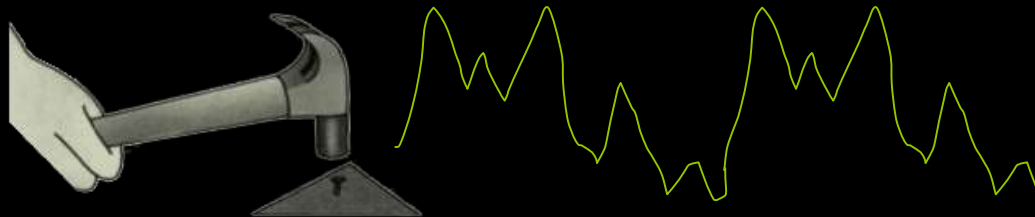
Introdução

- ▶ As vibrações no ar **se propagam como uma onda**, chamada onda de som

Onda de som
senoidal

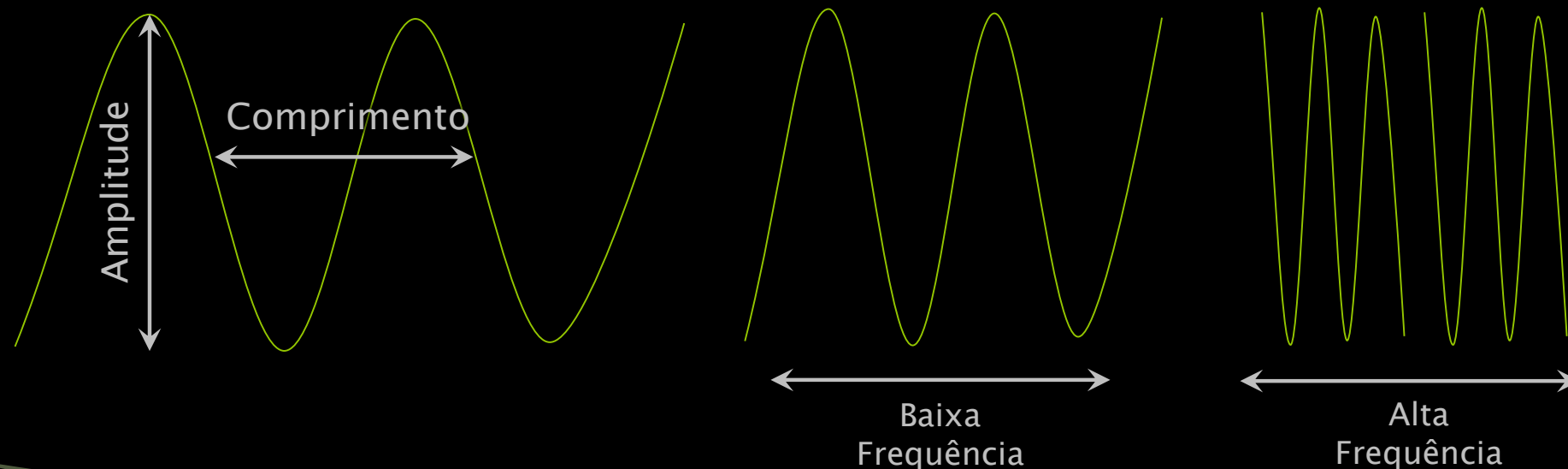


Onda de som
serrilhada



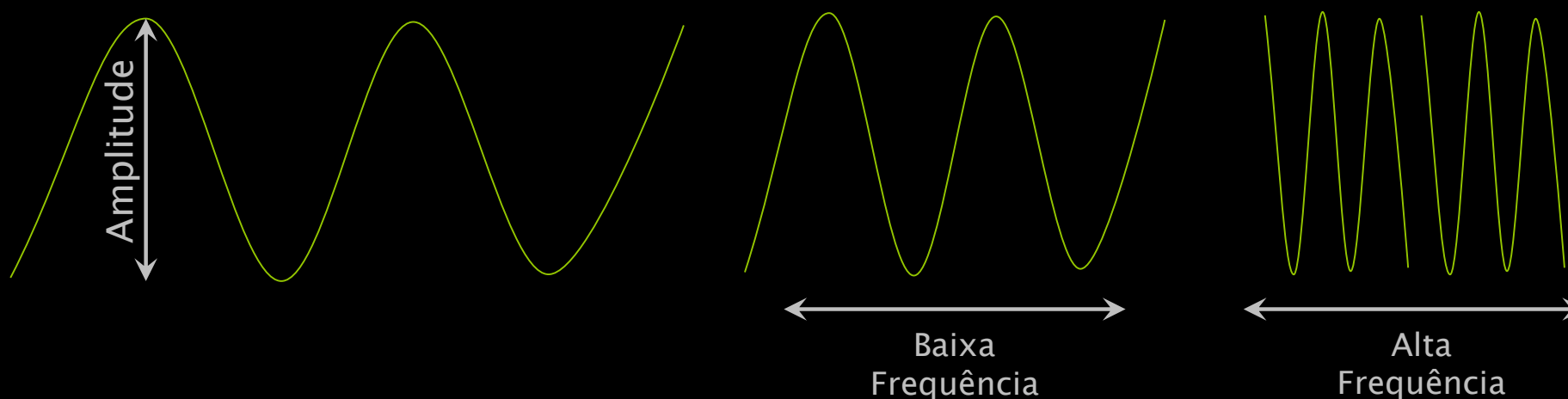
Introdução

- ▶ Uma onda de som é **caracterizada por**:
 - Amplitude: a força (ou altura) da onda
 - Comprimento: a distância entre dois pontos equivalentes
 - Frequência: número de vibrações por segundo



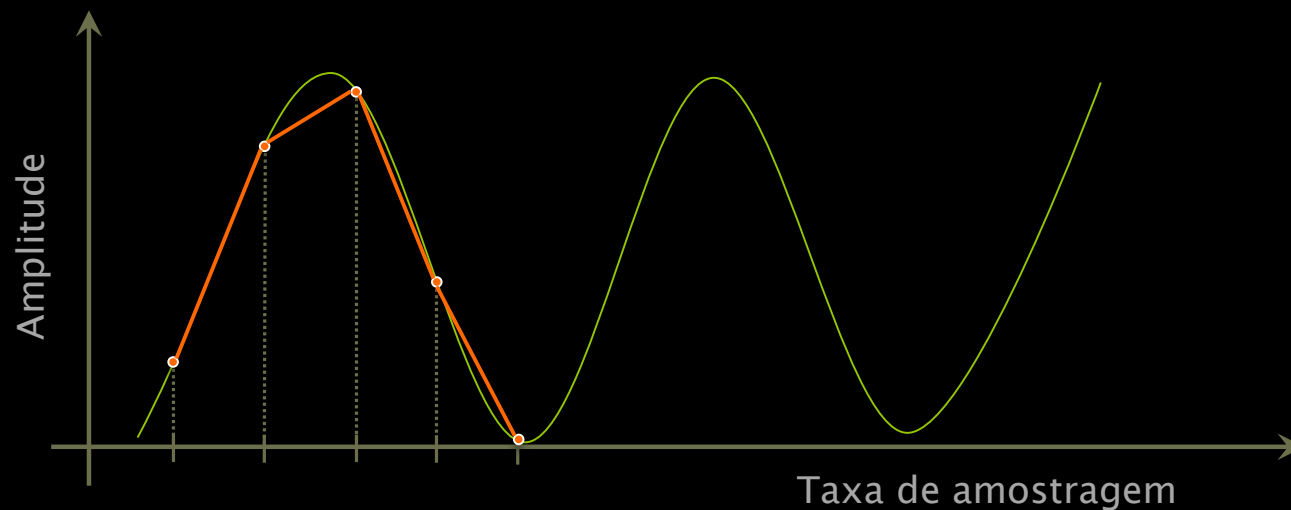
Representação Digital

- ▶ Para a **representação digital** utiliza-se:
 - Amplitude: se traduz diretamente em volume, quanto maior a amplitude maior o volume do som
 - Frequência: o ouvido humano é capaz de distinguir sons com frequências que variam de 20Hz a 20000Hz



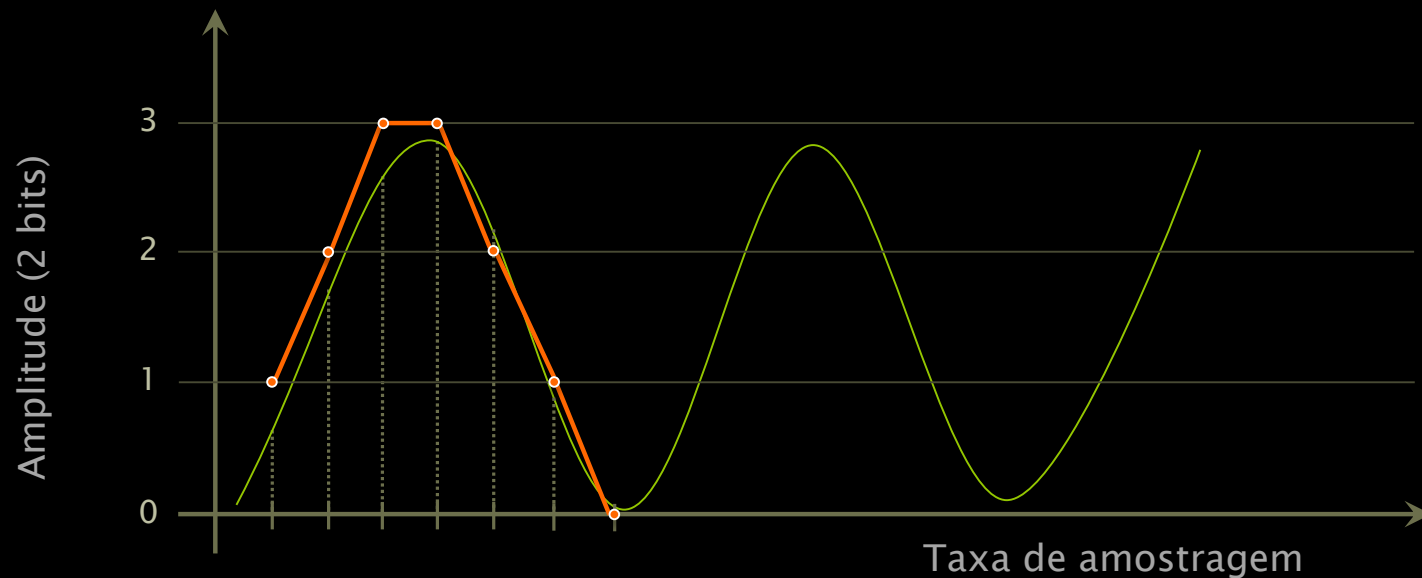
Representação Digital

- ▶ Um arquivo de áudio guarda **informações discretas** sobre a onda de som:
 - **Taxa de amostragem:** quanto maior a taxa, mais fiel será a onda armazenada. Ex.: 11025, 22050, 44100, 48000, 96000, 192000 Hz



Representação Digital

- ▶ Um arquivo de áudio guarda **informações discretas** sobre a onda de som:
 - **Bits**: quanto mais bits forem usados, mais fiel será a onda armazenada. Ex.: 8, 16, 24 bits



Representação Digital

- ▶ A tabela abaixo mostra taxas de amostragem comumente utilizadas e suas aplicações

| Amostragem | Uso |
|------------|--------------------------------------|
| 8000 Hz | Telefones (adequado para voz humana) |
| 11025 Hz | Áudio de vídeos de baixa qualidade |
| 22050 Hz | Efeitos sonoros em jogos |
| 44100 Hz | Qualidade de CD |
| 48000 Hz | Áudio de vídeos em formato DVD |
| 96000 Hz | Áudio de Blue-Ray e HD DVD |
| 192000 Hz | Áudio de Blue-Ray e HD DVD |

Áudio em Jogos

- ▶ Músicas e efeitos sonoros são elementos fundamentais para **colocar o jogador dentro de um universo virtual**
- Elementos sonoros são comumente usados:
 - **Fornecer tempo**
Ex.: urgência, tranquilidade, morte iminente
 - **Passar emoção**
Ex.: romantismo, perda, sofrimento, expectativa
 - **Passar autenticidade**
Ex.: efeitos sonoros (porta, passos, tiro, etc.)
 - **Fornecer um retorno sonoro**
Ex.: click de ativação ou seleção, pneu derrapando

Áudio em Jogos

▶ O **DirectX** fornece várias **soluções de áudio**:

- XAudio2: um motor para mixagem e processamento de áudio
- XACT*: usada para criar conteúdo
- X3DAudio*: usada para posicionar som 3D
- XAPO*: usada para criar efeitos sonoros

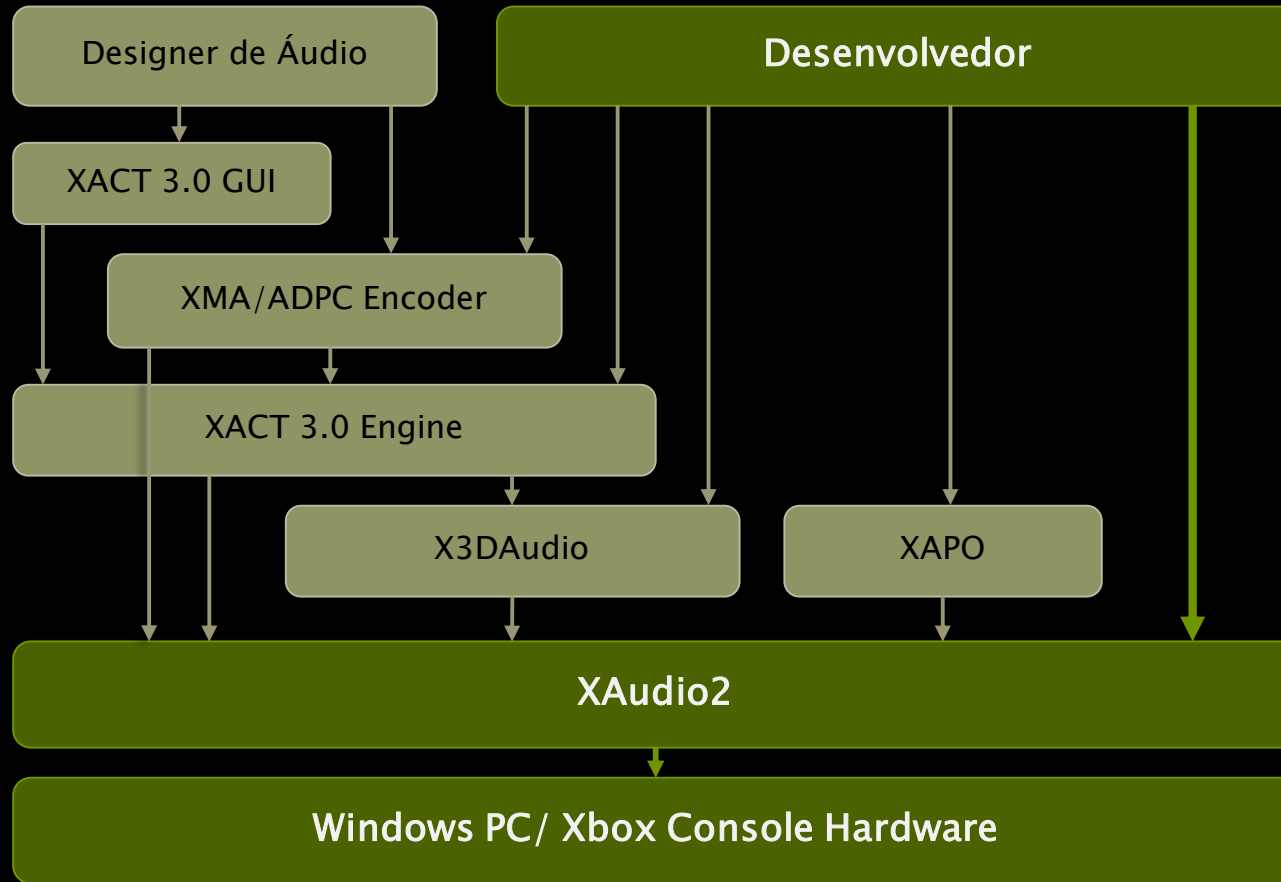
* Utilizam XAudio2 para tarefas de baixo nível

▶ A biblioteca **XAudio2**

- É a solução indicada para jogos
- Funciona no Windows e Xbox

Xaudio2 é uma API mais moderna que veio para substituir DirectSound e Xaudio

XAudio 2



Vamos usar diretamente a camada de mais baixo nível para construir nosso componente de áudio.

XAudio 2

► Características da API XAudio2

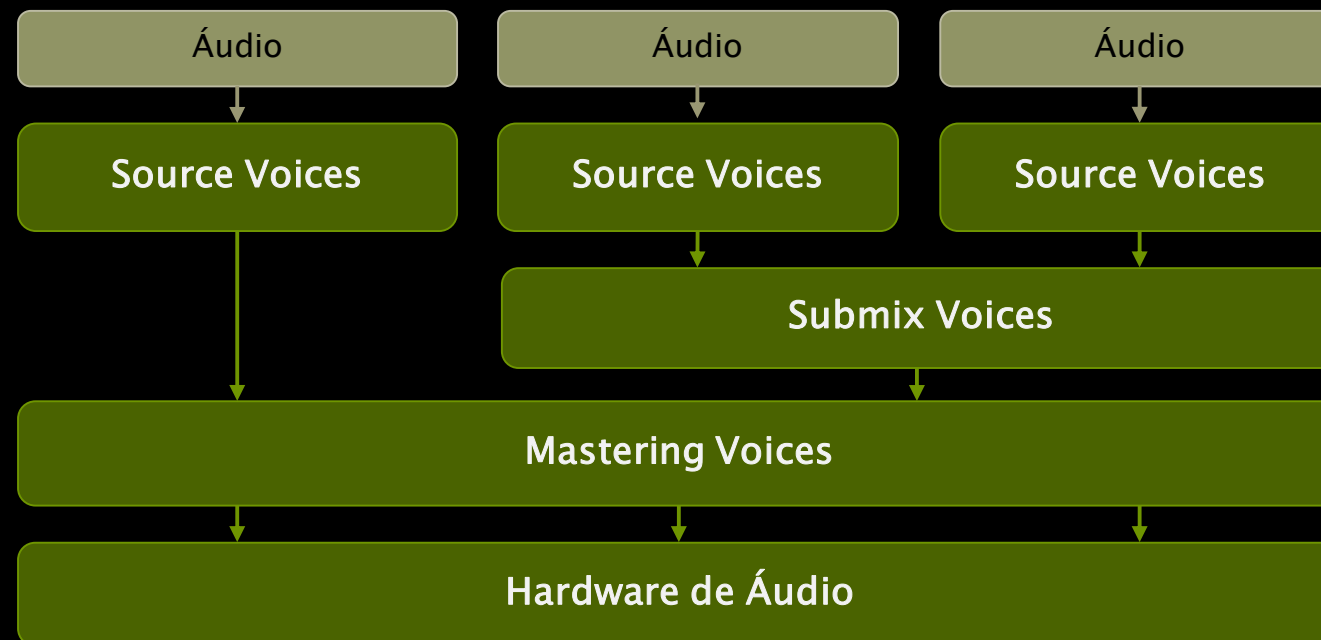
- **Transformações:** aplica efeitos de Processamento Digital de Sinais (DSP) e filtragem no som
- **Mixagem:** combina várias fontes diferentes de áudio em um único fluxo de som
- **Suporte a áudio comprimido:** ADPCM (Windows), XMA (Xbox) e xWMA (Windows e Xbox)
- Suporte a múltiplos canais e som surround
- Modelo de **API não bloqueante**

XAudio2

- ▶ XAudio2 trabalha com dois conceitos:
 - **Voices** são os objetos usados para representar o áudio
 - **Audio Graph** é uma coleção de Voices
- ▶ Existem vários tipos de Voices:
 - **Source Voices**: representam os dados de áudio
 - **Submix Voices**: fazem manipulações no áudio
 - **Mastering Voices**: recebem os dados de Source Voices ou Submix Voices e os enviam para o hardware de áudio

XAudio2

- ▶ **Audio Graph:** o áudio inicia em uma Source Voice, opcionalmente passa por uma ou mais Submix Voices e é enviado para uma Mastering Voice



Inicializando XAudio2

► Criar uma instância da engine XAudio2

```
IXAudio2 * audioEngine = nullptr;  
XAudio2Create(&audioEngine, 0, XAUDIO2_DEFAULT_PROCESSOR);
```

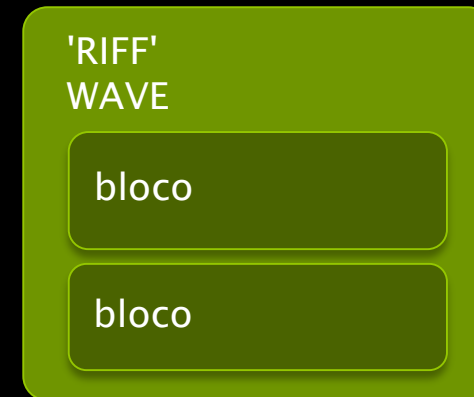
► Criar uma Mastering Voice:

```
IXAudio2MasteringVoice * masterVoice = nullptr;
```

```
audioEngine->CreateMasteringVoice(  
    &masterVoice,           // endereço do objeto  
    AUDIO2_DEFAULT_CHANNELS, // número de canais do sistema  
    XAUDIO2_DEFAULT_SAMPLERATE, // taxa de amostragem padrão (44100)  
    0,                      // sempre zero  
    0,                      // índice do dispositivo de áudio  
    NULL);                  // cascata de efeitos a serem usados
```

Formato de Dados RIFF

- ▶ **Resource Interchange File Format (RIFF)** é um formato de arquivo usado para armazenar dados em blocos
 - É o formato usado pelos **arquivos .wav**
 - O primeiro bloco, chamado '**RIFF**', contém o tipo do arquivo (WAVE para .wav) nos primeiros 4 bytes e os demais blocos no resto da sua seção de dados



Formato de Dados RIFF

- ▶ O tipo de dado armazenado em cada bloco é indicado por um código de 4 caracteres
 - **'fmt '**: contém o cabeçalho do arquivo que deve ser carregado em um registro WAVEFORMATEXTENSIBLE (.wav)
 - **'data'**: contém os dados do áudio que devem ser carregados em um registro XAUDIO2_BUFFER e passados para uma Source Voice

'RIFF'
WAVE

'fmt '
(cabeçalho)

'data'
(dados de áudio)

Carregando Arquivo WAVE

► Carregar o arquivo de áudio

```
// abre o arquivo de áudio para leitura
string fileName = "Resources/Intro.wav";
HANDLE hFile = CreateFile(fileName.c_str(), GENERIC_READ, FILE_SHARE_READ,
                          NULL, OPEN_EXISTING, 0, NULL);
```

► Copiar as seções fmt e data do arquivo:

```
// localiza e copia o bloco 'fmt' para WAVEFORMATEXTENSIBLE
FindChunk(hFile, fourccFMT, dwChunkSize, dwChunkPosition);
ReadChunkData(hFile, &wfx, dwChunkSize, dwChunkPosition);

// localiza e copia o bloco 'data' para XAUDIO2 BUFFER
FindChunk(hFile, fourccDATA, dwChunkSize, dwChunkPosition);
BYTE * pDataBuffer = new BYTE[dwChunkSize];
ReadChunkData(hFile, pDataBuffer, dwChunkSize, dwChunkPosition);
```

Tocando Áudio

► Inicializar o registro XAUDIO2_BUFFER

```
// preenche o registro XAUDIO2_BUFFER  
buffer.AudioBytes = dwChunkSize; // tamanho do buffer em bytes  
buffer.pAudioData = pDataBuffer; // dados (bits) do áudio  
buffer.Flags = XAUDIO2_END_OF_STREAM; // buffer único
```

► Criar Source Voice:

```
audioEngine->CreateSourceVoice(&sourceVoice, (WAVEFORMATEX*) &wfx);  
sourceVoice->SubmitSourceBuffer(&buffer);
```

► Tocar áudio:

```
sourceVoice->Start();
```

Resumo

- ▶ Música e efeitos sonoros são elementos importantes para **fornecer autenticidade** ao **mundo virtual** do jogo
- ▶ O DirectX fornece a biblioteca **XAudio2**
 - Ideal para desenvolvedores de jogos
 - Permite construir um **sistema de áudio**
 - Faz **mixagem** de diversas fontes
 - Pode ser usado tanto no Windows como no Xbox
 - Trabalha com os **arquivos .wav**