

# Controle do Áudio

Programação de Jogos

Judson Santos Santiago

# Introdução

- ▶ **Músicas e efeitos sonoros** são importantes
  - Fornecem tempo, emoção, autenticidade e retorno sonoro
- ▶ O DirectX fornece a biblioteca **XAudio2**
  - Permite construir um **sistema de áudio**
  - Faz **mixagem** de diversas fontes
  - Permite aplicação de efeitos
  - Pode ser usada tanto no Windows como no Xbox
  - Trabalha com o **formato de áudio WAVE**

# Introdução

► Para **tocar um som** usando XAudio2 é preciso:

- Carregar um arquivo .wav para a memória
  - Um **registro XAUDIO2\_BUFFER** guarda:
    - O tamanho do áudio em bytes
    - O endereço de memória com os dados do áudio
    - Flags de configuração

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

'RIFF'  
WAVE

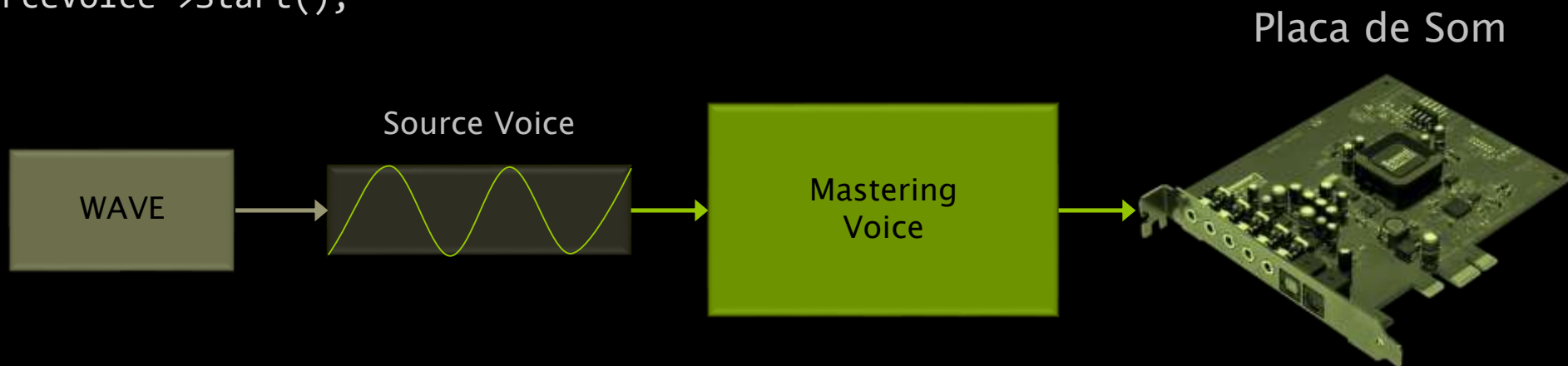
'fmt'  
(cabeçalho)

'data'  
(dados de áudio)

# Introdução

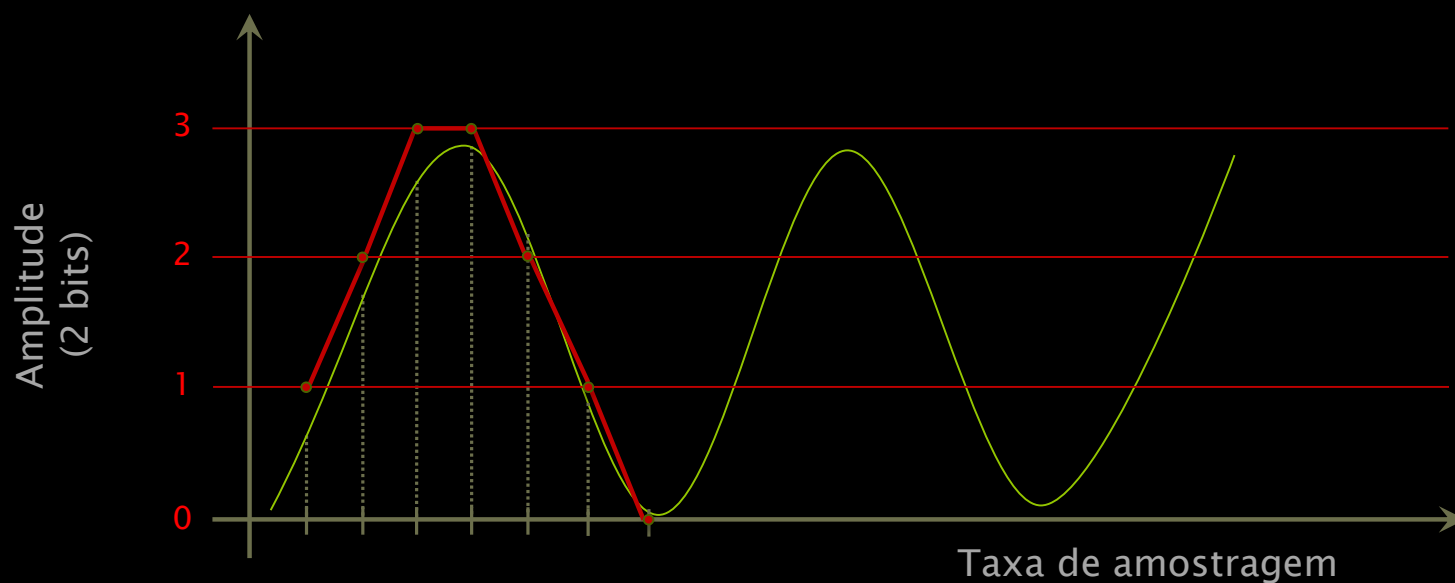
- ▶ Para **tocar um som** usando XAudio2 é preciso:
  - Criar uma **Source Voice** e enviar o áudio para reprodução

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



# Source Voices

- ▶ Cada Source Voice tem um **formato**
  - Definido pela quantidade de:
    - Bits por amostra (Ex.: 8 bits, 16 bits, etc.)
    - Amostras por segundo (Ex.: 22050 Hz, 44100 Hz, etc.)

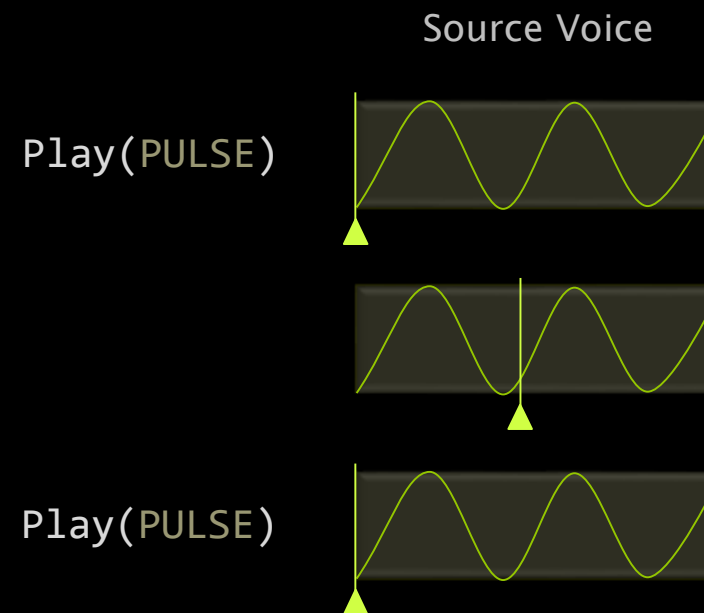


# Source Voices

► As Sources Voices funcionam como **trilhas de áudio**

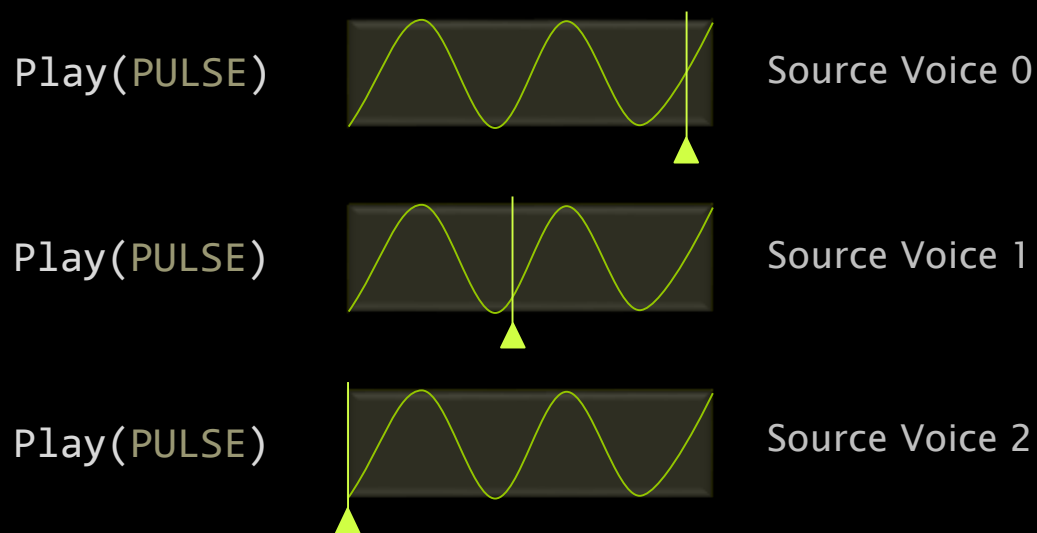
- Cada trilha toca um som por vez
- Um nova reprodução interrompe a reprodução atual:

```
// método Play()  
sourceVoice->Stop();  
sourceVoice->FlushSourceBuffers();  
sourceVoice->SubmitSourceBuffer(&buffer);  
sourceVoice->Start();
```



# Source Voices

- ▶ Para que um som possa ser reproduzido várias vezes **sem interromper o anterior** eles precisam usar Source Voices diferentes



Várias cópias do mesmo som sendo reproduzidos simultaneamente

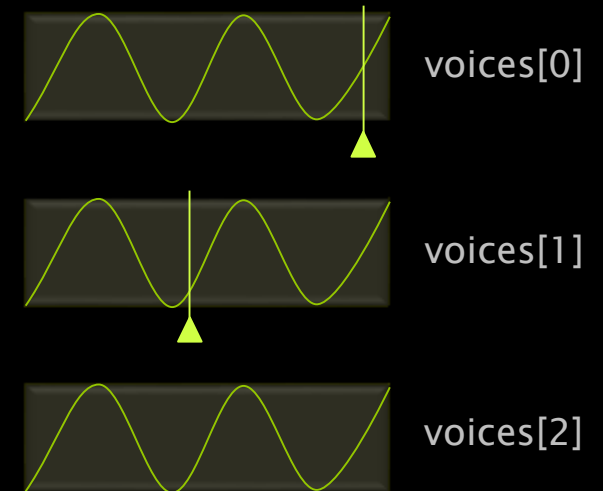
# Implementação

- ▶ Para implementar essa funcionalidade:
  - Cada som precisa de um **vetor** de Source Voices
  - O vetor deve ser usado de forma **circular**

```
void Audio::Play(uint id)
{
    // recupera som da tabela
    Sound * sound = sound_table[id];

    // limpa voz corrente e inicia reprodução do som
    sound->voices[sound->index]->Stop();
    sound->voices[sound->index]->FlushSourceBuffers();
    sound->voices[sound->index]->SubmitSourceBuffer(&sound->buffer);
    sound->voices[sound->index]->Start();

    // seleciona nova trilha para reprodução do próximo som
    sound->index = (sound->index + 1) % sound->tracks;
}
```

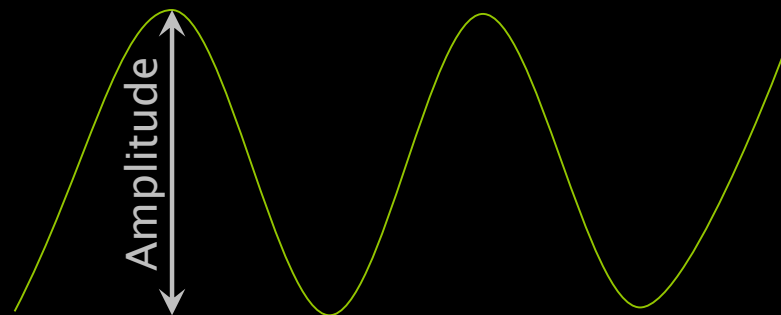




# Volume

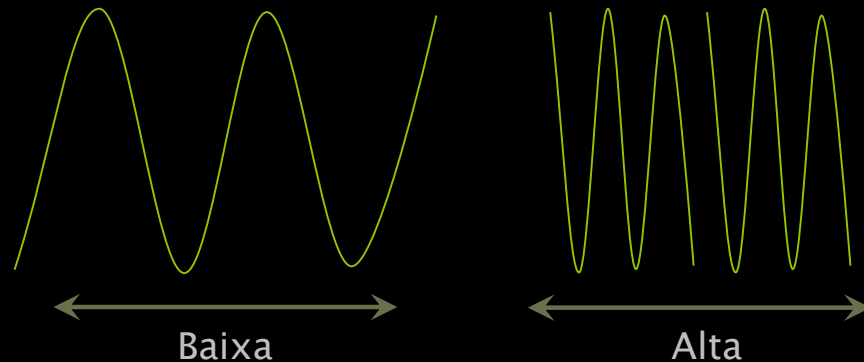
- ▶ O **volume** é definido pela **amplitude** da onda
  - É codificada por **números** em um arquivo de som
    - Pode ser modificada multiplicando o valor original por um fator
      - $2.0 * \text{amplitude}$  = duplica o volume
      - $0.5 * \text{amplitude}$  = reduz pela metade
  - Xaudio2 permite regular o volume através do método SetVolume:

```
sourceVoice->SetVolume(level);
```



# Frequência

- ▶ A **frequência do som** é definida pela quantidade de vezes que a onda se repete a cada segundo



A frequência também pode ser alterada através de um multiplicador

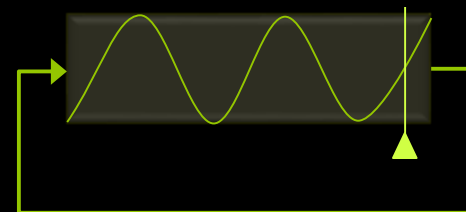
- Xaudio2 regula a frequência através do método SetFrequencyRatio:

```
sourceVoice->SetFrequencyRatio(level);
```

# Repetição

- ▶ Muitas vezes se deseja **tocar o som indefinidamente**
  - Xaudio2 permite tocar o som em **um laço infinito**
    - Basta ajustar um valor do registro XAUDIO2\_BUFFER
    - Deve ser feito antes de submeter o buffer para reprodução

```
XAUDIO2_BUFFER buffer;  
buffer.AudioBytes = dwChunkSize;      // tamanho do buffer em bytes  
buffer.pAudioData = pDataBuffer;      // dados (bits) do áudio  
buffer.Flags = XAUDIO2_END_OF_STREAM; // buffer único  
  
// toca áudio em loop ou apenas uma vez  
if (repeat)  
    sound->buffer.LoopCount = XAUDIO2_LOOP_INFINITE;  
else  
    ssound->buffer.LoopCount = 0;
```



# Resumo

- ▶ Biblioteca **XAudio2** fornece a possibilidade de controlar a reprodução do áudio além das funcionalidades básicas de Play e Stop
  - É possível alterar algumas características do som:
    - Volume
    - Frequência
  - O som pode ser tocado:
    - Em Sources Voices diferentes
    - Repetidamente em um laço infinito