XNA Sound

305890
Spring 2011
5/16/2011
Kyoung Shin Park
kpark@dankook.ac.kr

Sound

- SoundEffect/SoundEffectInstance class
- Song/MediaLibrary/MediaPlayer class
- Microsoft XACT (Cross-Platform Audio Creation Tool) Audio Authoring

SoundEffect

- SoundEffect/SoundEffectInstance
  - To play a sound without using XACT, you will use SoundEffect or SoundEffectInstance class
  - SoundEffect and SoundEffectInstance will be the only option for Zune games
  - A SoundEffect contains the audio data and metadata (such as wave data and loop information) loaded from a sound file.
  - You can create a SoundEffect by calling ContentManager.Load.

SoundEffectInstance

- SoundEffect/SoundEffectInstance
  - You can create multiple SoundEffectInstance objects and play them from a single SoundEffect.
  - You can create a SoundEffectInstance by calling SoundEffect.Play. SoundEffectInstance is created as playing, but you can pause it or stop it by calling Pause or Stop.
  - You can modify the volume, panning, and pitch of the SoundEffectInstance by setting the Volume, Pitch, and Pan properties or by creating the instance with one of the SoundEffect.Play methods that enables a user to specify the desired properties.

Playing a Sound using SoundEffect

- Declare SoundEffect

```
SoundEffect soundKaboom;
string soundName = “kaboom”;      // .wav sound
```

- Play the sound

```
soundKaboom = Content.Load<SoundEffect>(soundName );
soundKaboom.Play();                     // sound play

soundKaboom.Play(0.5f, 0.0f, 0.0f, true); // loop sound play
```

Playing a Sound using SoundEffect & SoundEffectInstance

- Declare SoundEffect & SoundEffectInstance

```
SoundEffect soundEngine;
SoundEffectInstance soundEngineInstance = null;
```

- Play the sound

```
soundEngine = Content.Load<SoundEffect>(“engine_2” );
soundEngineInstance = soundEngine.Play();
soundEngineInstance.Volume += 0.1f;

// set the pan speed so a full pan (-1.0 to 1.0 = 2.0)
// is achieved over the duration of the sound
panSpeed = 2.0f / (float)soundEngine.Duration.TotalSeconds;
// randomly choose panning direction
panSpeed = rnd.NextDouble() > 0.5 ? panSpeed : -panSpeed;
soundEngineInstance.Pan = panSpeed;
```

MediaLibrary

- MediaLibrary
  - **MediaLibrary** provides the following properties that return media collections: Albums, Artists, Genres, Pictures, Playlists, and Songs.
  - **On Windows, MediaLibrary cannot find any songs unless the Windows Media Player previously found songs on the system.** That means that Windows Media Player must first search the system for music before any songs can be accessed through MediaLibrary.

MediaPlayer

- MediaPlayer (static class)
  - **MediaPlayer** provides methods and properties for playing songs in the media library.
  - To control song playback, use the Play, Pause, Stop and Resume methods. MoveNext and MovePrevious methods move to the next or previous song in the queue.

Song

- Song
  - **Song** provides information about a song, including the song's Name, Artists, Album.
  - You can obtain a Song object through the SongCollection.Item indexer and the MediaQueue.ActiveSong property.
  - Before XNA 3.0, the only way to add audio to your game was to use XACT. All audio files had to be .wav files, which can be extremely large. Now, with XNA 3.0, they've provided another way that allows us to use .mp3 files and .wma files.

Playing a Song using MediaLibrary

- Declare MediaLibrary

```
MediaLibrary library = new MediaLibrary();
```

- Play the song

```
MediaPlayer.Stop();           // stop current audio playback
SongCollection songs = library.Songs;
Song song = songs[0];         // select the first song in the list
MediaPlayer.Play(song);       // play the song

MediaPlayer.Play(library.Albums[0].Songs[0]); // play a song
```

Playing a Song

- Declare Song

```
Song bgm;
```

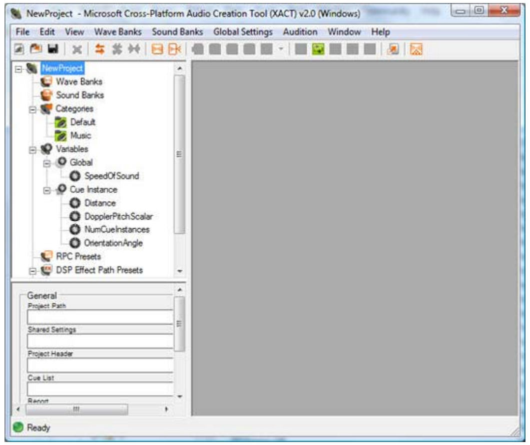
- Play the Sound

```
bgm = Content.Load<SoundEffect>("엘린+숲");
MediaPlayer.IsRepeating = true;     // looping sound
MediaPlayer.Volume = 0.5f;          // set volume (0.0 ~ 1.0)
MediaPlayer.Play(bgm);       // play a background music

MediaPlayer.Stop();          // stop the music
```

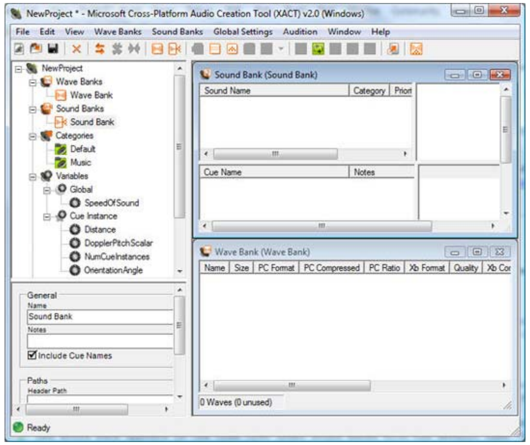
XACT

- XNA Game Studio 3.0 => Tools => Microsoft Cross Platform Audio Creation Tool (XACT)



Adding Sounds to XACT

- Create a new XACT project (e.g., simple.xap)

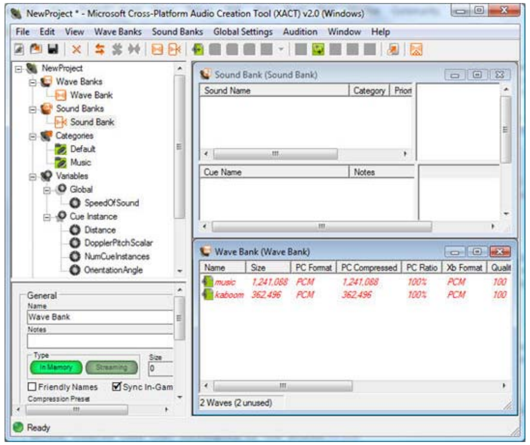


General
Name: Sound Bank
Notes:
☑ Include Cue Names

0 Waves (0 unused)

Adding Sounds to XACT

- Add sound files in Wave Bank



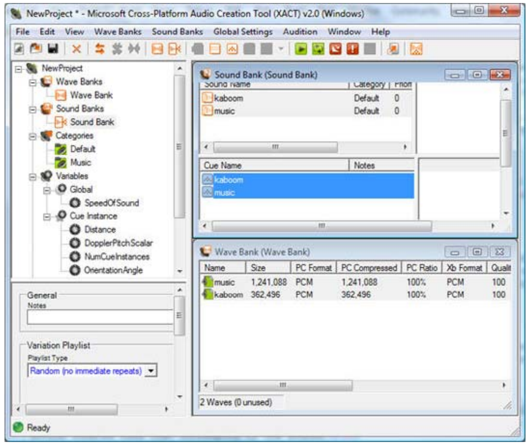
General
Name: Wave Bank
Notes:
Type: In Memory / Streaming
Size: 0
☐ Friendly Names ☑ Sync In-Gam

Name	Size	PC Format	PC Compressed	PC Ratio	Xb Format	Quali
music	1,241,088	PCM	1,241,088	100%	PCM	100
kaboom	362,496	PCM	362,496	100%	PCM	100

2 Waves (2 unused)

Adding Sounds to XACT

- Drag the wave from the Wave Bank to the Sound Bank
- Then drop the wave on top of the Cue Name panel



Sound Name	Category	Priori
kaboom	Default	0
music	Default	0

Cue Name	Notes
kaboom	
music	

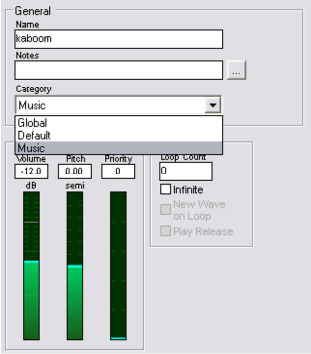
Name	Size	PC Format	PC Compressed	PC Ratio	Xb Format	Quali
music	1,241,088	PCM	1,241,088	100%	PCM	100
kaboom	362,496	PCM	362,496	100%	PCM	100

2 Waves (0 unused)

General
Notes:
Variation Playlist
Playlist Type: Random (no immediate repeats)

Adding Sounds to XACT

- To specify background music, select "Music" in the category drop down box, and save the project



General
Name: kaboom
Notes:
Category: Music
Volume: -12.0
Pitch: 0.00
Priority: 0
Loop Count: 0
☐ Infinite
☐ New Wave on Loop
☐ Play Release

Adding Sounds to XACT

- To save the project, File->Save Project
- To build the project, File->Build

Playing Sounds using XACT

- Load the XACT project (e.g., simple.xap) into the Content
- Declare AudioEngine, SoundBank, WaveBank

```
// Audio objects
AudioEngine engine;
SoundBank soundBank;
WaveBank waveBank;
engine = new AudioEngine("Content/Audio/simple.xgs");
soundBank = new SoundBank(engine, "Content/Audio/SimpleSoundBank.xsb");
waveBank = new WaveBank(engine, "Content/Audio/SimpleWaveBank.xwb");
```

- Update audio

```
audioEngine.Update();
```

- Play sound cues

```
soundBank.PlayCue(”kaboom”);
soundBank.PlayCue(”music”);
```