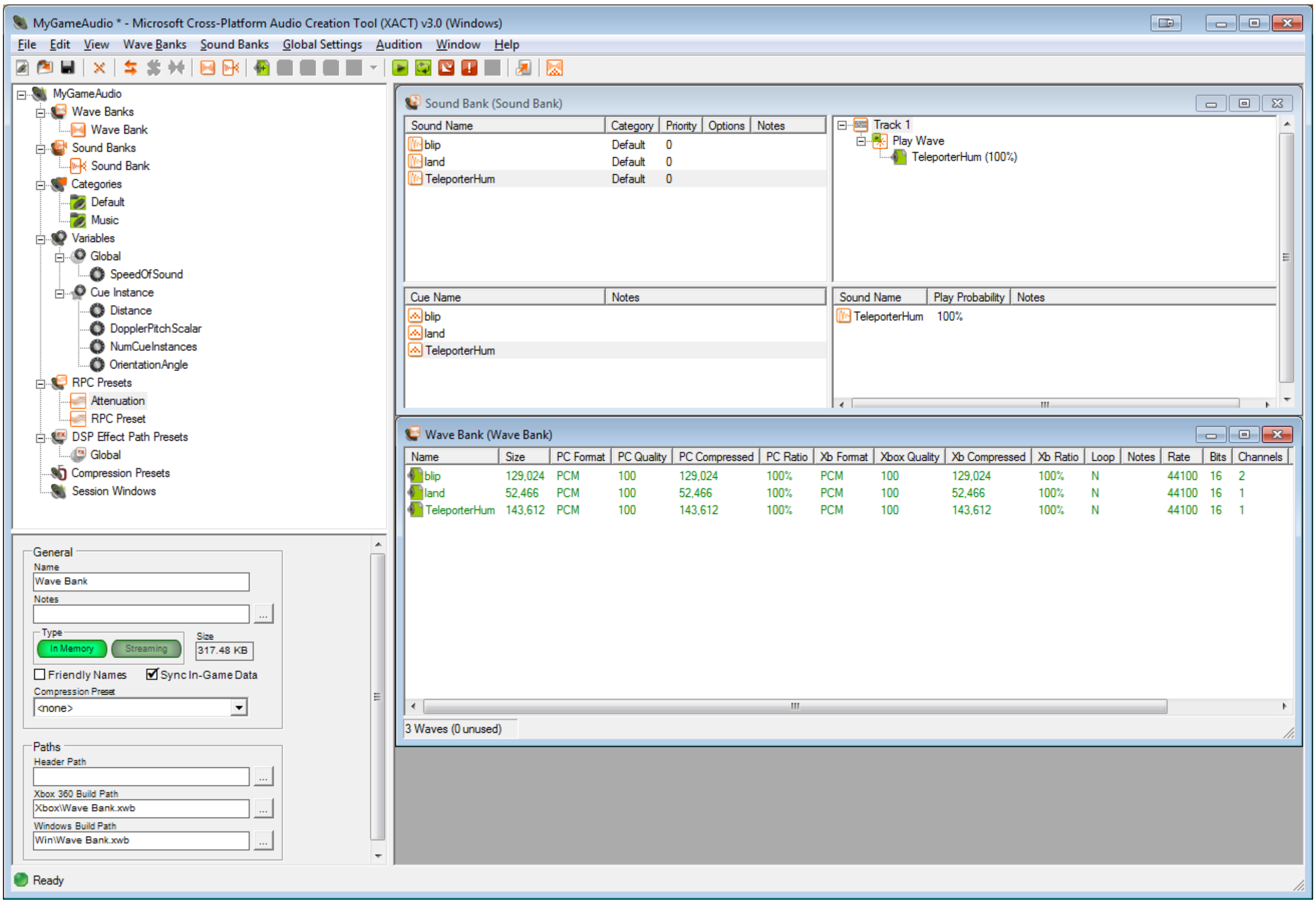
Sound Workflow

by Patrick O Halloran (https://mahara.dkit.ie/user/view.php?id=628)

Creating the Sound Engine

To create a sound engine for XNA, a seperate software package is used called XACT (Cross-platform Audio Creation Tool). XACT is also used with the DirectX SDK. Its purpose is to seperate the tasks of programming and sound engineering. The sound engineer can tweak sound behaviour in XACT without having to worry about the code implementation.

Click to enlarge



Sound Name	Category	Priority	Options	Notes
blip	Default	0		
land	Default	0		
TeleporterHum	Default	0		

Cue Name	Notes
blip	
land	
TeleporterHum	

Sound Name	Play Probability	Notes
TeleporterHum	100%	

Name	Size	PC Format	PC Quality	PC Compressed	PC Ratio	Xb Format	Xbox Quality	Xb Compressed	Xb Ratio	Loop	Notes	Rate	Bits	Channels
blip	129,024	PCM	100	129,024	100%	PCM	100	129,024	100%	N		44100	16	2
land	52,466	PCM	100	52,466	100%	PCM	100	52,466	100%	N		44100	16	1
TeleporterHum	143,612	PCM	100	143,612	100%	PCM	100	143,612	100%	N		44100	16	1

3 Waves (0 unused)

General
Name: Wave Bank
Notes:
Type: In Memory / Streaming
Size: 317.48 KB
☐ Friendly Names ☑ Sync In-Game Data
Compression Preset: <none>

Paths
Header Path:
Xbox 360 Build Path: Xbox\Wave Bank.xwb
Windows Build Path: Win\Wave Bank.xwb

(https://mahara.dkit.ie/artefact/artefact.php?artefact=44448&view=10391)

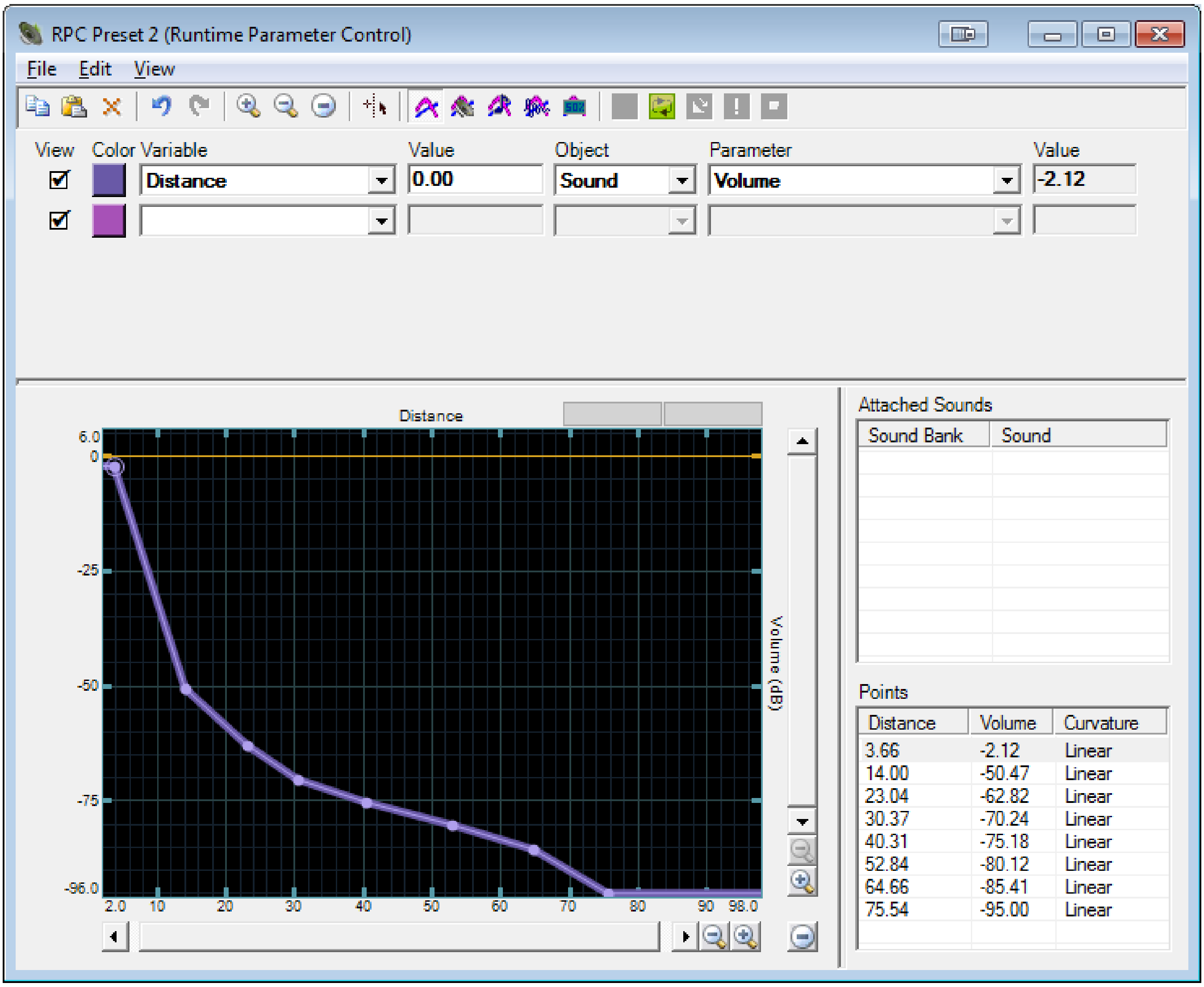
Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44448)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44448&view=10391&block=57643)

To create the Audio Engine, you need to create a new project file in XACT (.xap). As you can see in the screenshot above, I then added a Sound Bank and a Wave Bank, which are needed for the engine in XNA. I added .wav files to these from the GameContent folder in the XNA project to avoid any filepath issues.

When I implemented the engine in XNA, I discovered the the 3d sound wasn't working (The sounds were just playing globally). This is because I needed to set attenuation in XACT, to control how the sound falls off the further from the listener it is.

Attenuation



View	Color	Variable	Value	Object	Parameter	Value
☑	🟪	Distance	0.00	Sound	Volume	-2.12
☑	🟪					

Attached Sounds

Sound Bank	Sound

Points

Distance	Volume	Curvature
3.66	-2.12	Linear
14.00	-50.47	Linear
23.04	-62.82	Linear
30.37	-70.24	Linear
40.31	-75.18	Linear
52.84	-80.12	Linear
64.66	-85.41	Linear
75.54	-95.00	Linear

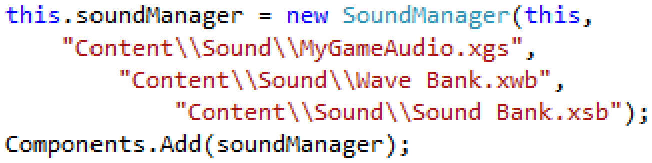
(https://mahara.dkit.ie/artefact/artefact.php?artefact=44453&view=10391)

➔ Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44453)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44453&view=10391&block=57651)

Implementing the Sound Engine

Implementing the AudioEngine functionality required a rewrite of the Soundmanager class we were using, to take in the audio engine, wave and sound banks, generated by the xap file.

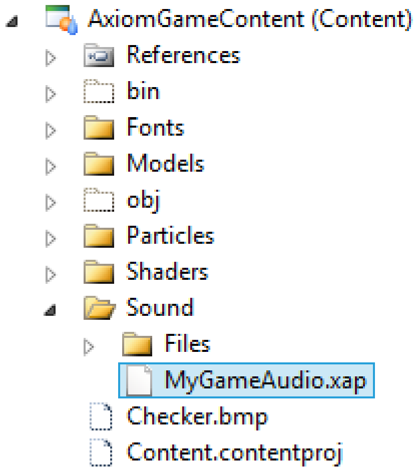


```
this.soundManager = new SoundManager(this,
    "Content\\Sound\\MyGameAudio.xgs",
        "Content\\Sound\\Wave Bank.xwb",
            "Content\\Sound\\Sound Bank.xsb");
Components.Add(soundManager);
```

(https://mahara.dkit.ie/artefact/artefact.php?artefact=44420&view=10391)

Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44420)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44420&view=10391&block=57622)



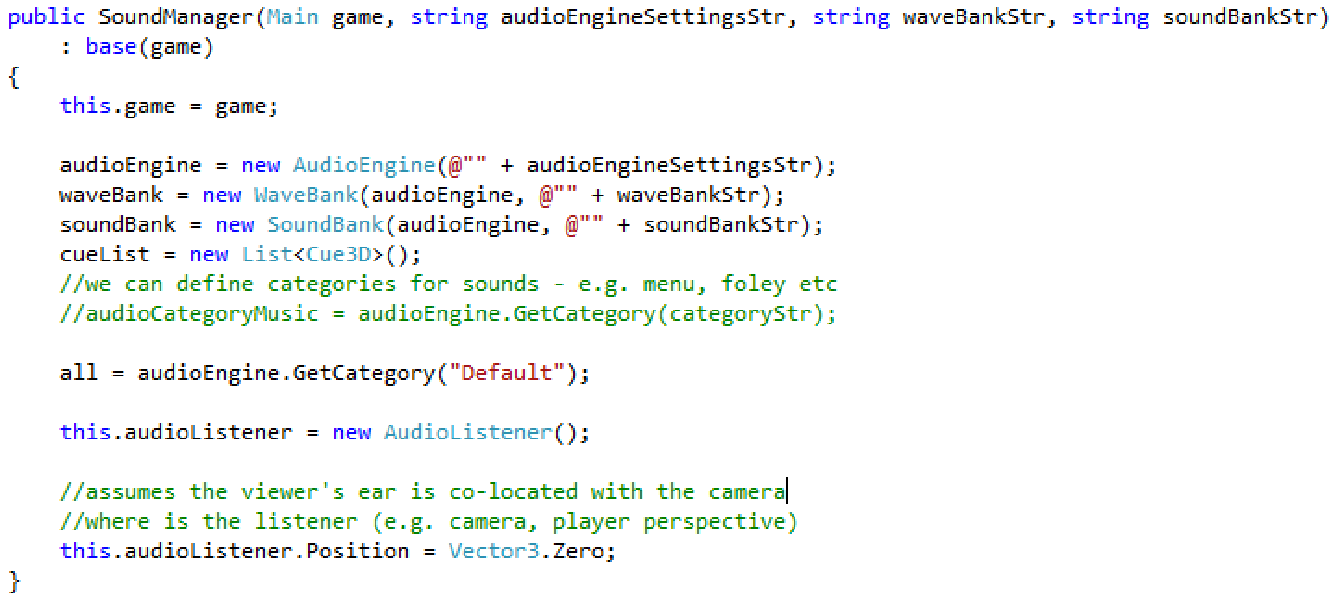
- AxiomGameContent (Content)
  - References
  - bin
  - Fonts
  - Models
  - obj
  - Particles
  - Shaders
  - Sound
    - Files
    - MyGameAudio.xap
  - Checker.bmp
  - Content.contentproj

(https://mahara.dkit.ie/artefact/artefact.php?artefact=44432&view=10391)

Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44432)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44432&view=10391&block=57629)

Click to enlarge



```
public SoundManager(Main game, string audioEngineSettingsStr, string waveBankStr, string soundBankStr)
    : base(game)
{
    this.game = game;

    audioEngine = new AudioEngine(@"" + audioEngineSettingsStr);
    waveBank = new WaveBank(audioEngine, @"" + waveBankStr);
    soundBank = new SoundBank(audioEngine, @"" + soundBankStr);
    cueList = new List<Cue3D>();
    //we can define categories for sounds - e.g. menu, foley etc
    //audioCategoryMusic = audioEngine.GetCategory(categoryStr);

    all = audioEngine.GetCategory("Default");

    this.audioListener = new AudioListener();

    //assumes the viewer's ear is co-located with the camera
    //where is the listener (e.g. camera, player perspective)
    this.audioListener.Position = Vector3.Zero;
}
```

(https://mahara.dkit.ie/artefact/artefact.php?artefact=44433&view=10391)

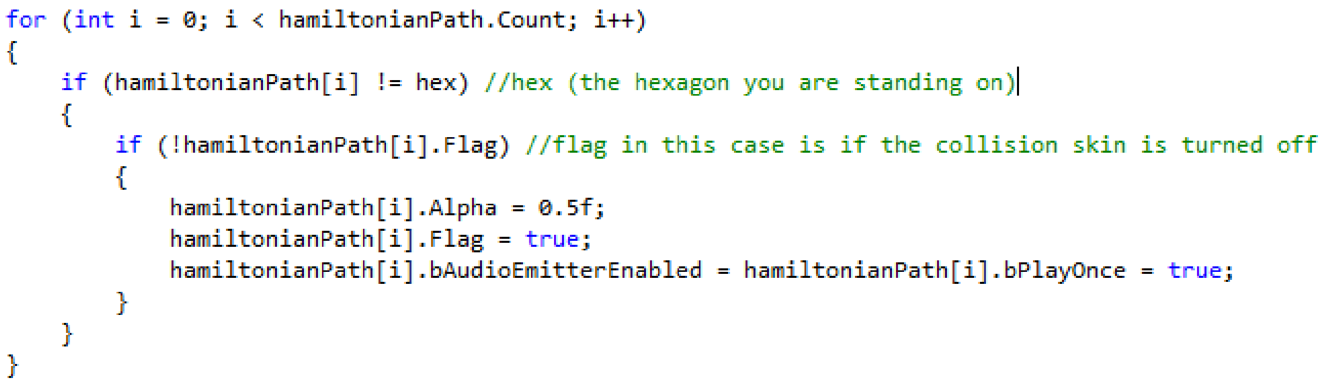
Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44433)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44433&view=10391&block=57631)

A cuelist is created, which a list of Cue3d objects. Cue3d is a wrapper class that contains a cue, which is the sound, and an audioEmitter, which controls where you hear that sound.

An AudioListener is created, which you update with the players position so 3d/environmental sounds can be achieved.

Example Usage



```
for (int i = 0; i < hamiltonianPath.Count; i++)
{
    if (hamiltonianPath[i] != hex) //hex (the hexagon you are standing on)
    {
        if (!hamiltonianPath[i].Flag) //flag in this case is if the collision skin is turned off
        {
            hamiltonianPath[i].Alpha = 0.5f;
            hamiltonianPath[i].Flag = true;
            hamiltonianPath[i].bAudioEmitterEnabled = hamiltonianPath[i].bPlayOnce = true;
        }
    }
}
```

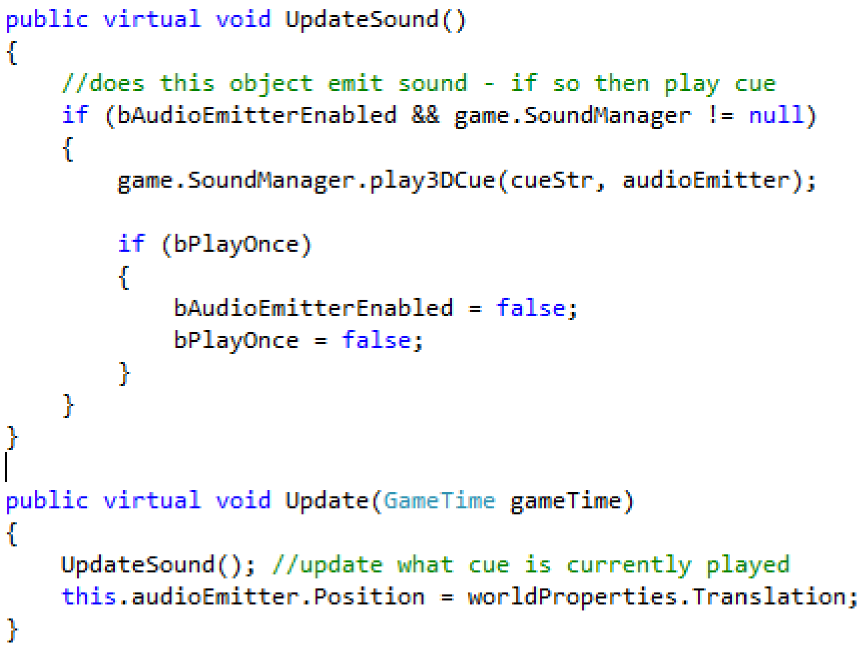
(https://mahara.dkit.ie/artefact/artefact.php?artefact=44483&view=10391)

Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44483)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44483&view=10391&block=57684)

In this example, whenever a hexagon is added to a list of those that are used, they make their "powering off" sound. They only make this sound once so you make bPlayOnce equal to true aswell as bAudioEmitterEnabled.

Updating the Sound Engine



```
public virtual void UpdateSound()
{
    //does this object emit sound - if so then play cue
    if (bAudioEmitterEnabled && game.SoundManager != null)
    {
        game.SoundManager.play3DCue(cueStr, audioEmitter);

        if (bPlayOnce)
        {
            bAudioEmitterEnabled = false;
            bPlayOnce = false;
        }
    }
}

public virtual void Update(GameTime gameTime)
{
    UpdateSound(); //update what cue is currently played
    this.audioEmitter.Position = worldProperties.Translation;
}
```

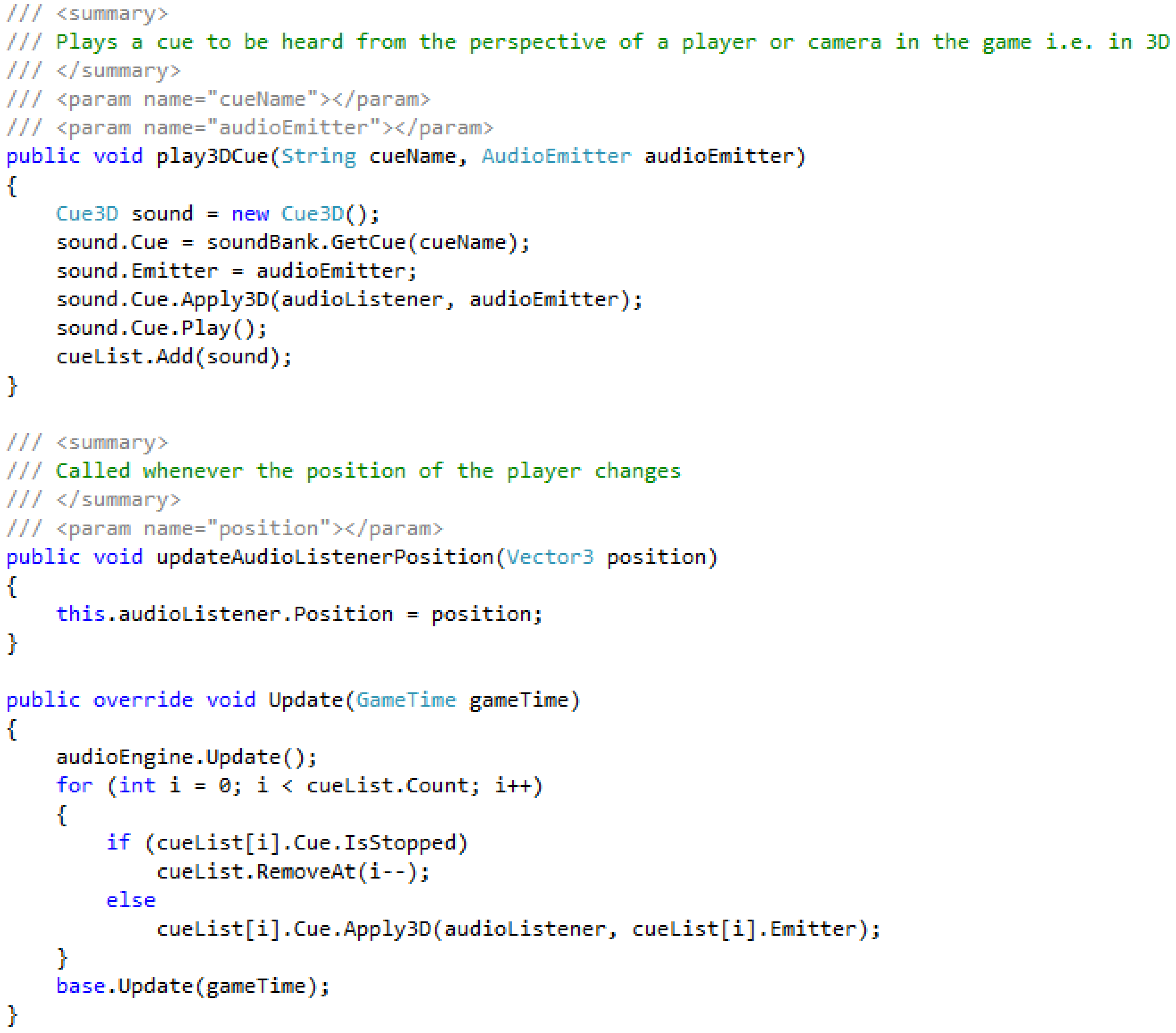
(https://mahara.dkit.ie/artefact/artefact.php?artefact=44434&view=10391)

Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44434)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44434&view=10391&block=57633)

On each update each object in the game will update it's audioEmitter position, and play its sound if it's emitter is enabled. If playonce is true, the emitter will be disabled so that the sound doesn't loop.

Click to enlarge



```
/// <summary>
/// Plays a cue to be heard from the perspective of a player or camera in the game i.e. in 3D
/// </summary>
/// <param name="cueName"></param>
/// <param name="audioEmitter"></param>
public void play3DCue(String cueName, AudioEmitter audioEmitter)
{
    Cue3D sound = new Cue3D();
    sound.Cue = soundBank.GetCue(cueName);
    sound.Emitter = audioEmitter;
    sound.Cue.Apply3D(audioListener, audioEmitter);
    sound.Cue.Play();
    cueList.Add(sound);
}

/// <summary>
/// Called whenever the position of the player changes
/// </summary>
/// <param name="position"></param>
public void updateAudioListenerPosition(Vector3 position)
{
    this.audioListener.Position = position;
}

public override void Update(GameTime gameTime)
{
    audioEngine.Update();
    for (int i = 0; i < cueList.Count; i++)
    {
        if (cueList[i].Cue.IsStopped)
            cueList.RemoveAt(i--);
        else
            cueList[i].Cue.Apply3D(audioListener, cueList[i].Emitter);
    }
    base.Update(gameTime);
}
```

(https://mahara.dkit.ie/artefact/artefact.php?artefact=44436&view=10391)

Add comment (https://mahara.dkit.ie/artefact/artefact.php?view=10391&artefact=44436)

Details (https://mahara.dkit.ie/artefact/artefact.php?artefact=44436&view=10391&block=57637)

When play3DCue is called, it gets the cue from the soundbank, begins playing it, and adds it to the cueList (currently playing cues). In the update, the 3d effect is updated by supplying the cue with the most up to date listener and emitter positions for each object. When the cue is finished it is removed from the list.

Comments

Add comment

Paragraph **B** *I*

p

Rating ⊘ ★ ★ ★ ★ ★

Make public Yes ○

Attach file

Choose File No file chosen

+ Add attachment ()

Comment Cancel

43 page visits from 1 March 2013 to 29 November 2016

powered by mahara (https://mahara.org/)

Privacy statement (https://mahara.dkit.ie/privacy.php) About (https://mahara.dkit.ie/about.php)

Contact us (https://mahara.dkit.ie/contact.php)