Sop HDR Audio Package for Unity 3D

sop3000@sopwerk.com
http://sop3000.sopwerk.com/hdraudio

This package is an implementation of the High Dynamic Range (HDR) audio manager for Unity 3D.
It is implemented according to the principles described in the DICE GDC'09 presentation "How High Dynamic Range Audio Makes Battlefield: Bad Company Go BOOM".

During the development of my sop3000 game, I've faced an issues with having too many sounds playing simultaneously in a single scene: I've hit the Unity limit on number of voices, sound priorities were difficult to setup and manage, ducking was hard to do properly, etc.

I've chosen to implement an HDR audio approach in my game, as it promised to do mixing and prioritizing of all sounds in the scene automatically, and to give me a simple way to setup and manage sound definitions.

I'm quite happy how it turned out: the setup is easy enough, tweaking of the sounds does not take too much time and resulting mix is really good, as far as I can tell. But of course, your result may wary - it depends on the type of game this approach is used for.

You can see how the audio manager actually works in the sop3000 game demo.

How To Use the Package

In order to better understand principles of the HDR audio, I would recommend to go through DICE GDC'09 presentation.

A good introduction into HDR audio can also be found in the Wwise HDR User's Guide.
It differs slightly from the way DICE did it, but I find the introduction is still good.

Sop HDR Audio Demo

I've put together a simple demo application to show basics of the package:

- Sop HDR Audio Demo (webplayer)

On the left panel of the demo application there is a single listener icon surrounded by multiple sound sources, all represented by different icons. The right panel shows main HDR audio properties and some properties of the currently selected sound source. You can start or stop any sound at any time by clicking on the corresponding icon. By dragging icons around on the screen, you can see how the distance between the sound source and listener affects Perceived Loudness and Volume properties of the sound.

Debug Window

The Debug Window (accessible from *Windows -> HDR Audio Debug Window* menu) provides detailed information about all active HDR sounds on the scene in the real time. It can be quite useful during the fine-tuning of HDR properties.

Setup in a new Scene

1. On the empty scene, create a new game object (e.g. HDRAudio) and assign `HDRAudio.cs` script to it. The HDRAudio service is a singleton and is accessible from everywhere in the code via `HDRAudio.Instance` property.

2. In order for the audio clip to be played by the HDRAudio service, you will have to create a **sound definition** prefab - just a regular game object with either `SingleSoundDef.cs` or `GroupSoundDef.cs` script assigned to it. The `SingleSoundDef.cs` definition is used by single audio clip sounds. The `GroupSoundDef.cs` script is used to define a sound that will play one randomly selected audio clip from a list on each invocation. See *Turret* sound prefab in `Sopwerk/HDRAudio/Demo/Sounds` folder for example of group sound definition.

3. Now the sound can be played as shown in the following example:

```
    using UnityEngine;
    using Sopwerk.HDRAudio;

    public class HowToUseHDRAudio : MonoBehaviour
    {
        // Should be linked to the Sound Definition prefab via the Unity
        // editor inspector window.
        [SerializeField]
        private SoundDef _soundDef = null;

        public void PlaySound()
        {
            HDRAudio.Instance.Play(_soundDef, transform.position, transform);
        }
    }
```

Sound Definition Properties

- The `Loudness` property is used by the audio manager to compute perceived loudness of a sound in the scene.

- The `Priority` property controls which sound will be culled when there are too many sound clips playing at once.

- Section `Near Player Mods` is used to reduce loudness of a sound played in close proximity to the listener. For instance, the *Thruster* sound in the demo application has `Near Player Loudness Mod` property defined. Try to drag *Rocket* icon over the listener icon to hear a reduction of the sound loudness by -10 (dB).

- The `Repeat Rate` property is not used directly by the audio manager but together with the RepeatingSoundLimiter utility class. See the `SoundSourceWidger.RepeatPlaySound()` method for usage example.

- The properties in the `Audio Source` section are basically the same as in the AudioSource unity component.

- The `RMS Curves` feature makes it possible for the audio manager to control a top position of the HDR window more precisely - based on an actual amplitude of the playing sound clip.
  The RMS curve can be generated and then edited directly in the Inspector window of the selected sound prefab. From my experience, it makes little sense to define RMS curve for loops or for sounds with even amplitude. But you can see an immediate positive effect on the explosion or gunshot sounds.

Credits

- HDR Audio: a dynamic mixing technique used in EA Digital Illusions CE Frostbite Engine to allow relatively louder sounds to drown out softer sounds.
- All icons are from http://game-icons.net:
  - Screen impact icon by Lorc
  - Turret icon by sbed
  - Vortex icon by Lorc
  - Alien stare icon by Lorc
  - Whirlwind icon by Lorc
  - Rocket icon by Lorc
  - Human ear icon by Delapouite
- Move cursor is from http://www.iconarchive.com by Oxygen Team
- Background texture is Brushed Aluminum texture from http://www.texturex.com
- Sound clips:
  - cgeffex_tornado.wav by CGEffex
  - quaker540_hq-explosion.wav by Quaker540
  - mickboere_jet-loop-01.wav by MickBoere
  - Answer Long1.waw, Answer Long2.wav, Answer Long3.wav, Excited Long1.wav, Question Long1.wav, Question Long2.wav are from Gibberish Dialogue & Voice pack Volume 1 free asset by Rocklynn Productions
  - Laser_00.wav, WarpDrive_01.wav are from Sci-Fi Sfx free asset by Little Robot Sound Factory

Contact

sop3000@sopwerk.com
http://sop3000.sopwerk.com/hdraudio

LICENSE

The license below covers all of the C# code files in this package, all non-code assets, including but not limited to: models, textures, sounds, etc. are the property of their respective author and under other licenses.
