

Introduction to Sound

Video

http://www.leadwerks.com/files/Tutorials/CPP/Introduction_To_Sound.wmv

Files

http://www.leadwerks.com/files/Tutorials/CPP/Introduction_To_Sound_Files.zip

Sound in Leadwerks Engine

Sound is second only to sight as our most important sensory experience. In games, sound subtly communicates information about our surroundings to us, like where we are and even the material our surroundings are made of. Sound is provided in Leadwerks Engine with OpenAL. The engine provides sound and source classes which are thin wrappers on top the OpenAL buffer and source objects.

It's easy to load and play a sound:

```
TSound sound=LoadSound("tone.wav");
PlaySound(sound);
```

For more advanced control over a sound, we need to call `CreateSource` and return a sound source. The source will not play until we tell it to, so we can adjust settings before it is audible:

```
TSound sound=LoadSound("tone.wav");
TSource source=CreateSource(sound);
PlaySource(source);
```

Let's listen to sounds in action. Start with this simple program which will load and play a sound:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(800,600);
```

```

//Create a world
if (!CreateWorld()) {
    MessageBoxA(0, "Error", "Failed to create world.", 0);
    goto exitapp;
}

//Create a camera
TEntity cam=CreateCamera();
CameraClearColor(cam, Vec4(0, 0, 1, 1));

TSound sound=LoadSound("tone.wav");
TSource source=CreateSource(sound);
PlaySound(source);

//Main loop
while(!KeyHit(KEY_ESCAPE)) {

    //Update the world
    UpdateWorld();

    //Render the world
    RenderWorld();

    //Swap the back and front buffers
    Flip();
}

exitapp:
return Terminate();
}

```

Controlling Sound

We can play, pause, resume, and stop any source at any time. When we play a source, the source rewinds to the beginning of the sound and starts playing again. Add this code inside the main loop and run the program. Every time you press the P key the sound will play from the beginning:

```

if (KeyHit(KEY_P)) {
    PlaySource(source);
}

```

To play a repeating sound, we can specify the SOURCE_LOOP flag when creating the source. Modify your code to match this:

```

TSource source=CreateSource(sound, SOURCE_LOOP);

```

We can pause, resume, and even get the current state of a source. When we resume a source that has been paused, the sound plays from the position we paused it at, instead of rewinding. Add this code inside the main loop and you will be able to pause and resume the source with the space key.

```
if (KeyHit(KEY_SPACE)) {
    if (GetSourceState(source)==SOURCE_PLAYING) {
        PauseSource(source);
    }
    else {
        ResumeSource(source);
    }
}
```

We can also adjust settings like pitch and volume. Declare this variable before the main loop:

```
float pitch=1.0;
```

Now add this code inside the main loop. You can now control the sound pitch with the + and – keys on your keyboard number pad:

```
if (KeyDown(KEY_NUMADD)) {
    pitch=pitch+0.001;
    SetSourcePitch(source,pitch);
}

if (KeyDown(KEY_NUMSUBTRACT)) {
    pitch=pitch-0.001;
    SetSourcePitch(source,pitch);
}
```

Here is our finished sound control code:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(800,600);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0,"Error","Failed to create world.",0);
        goto exitapp;
    }
}
```

```

//Create a camera
TEntity cam=CreateCamera();
CameraClearColor(cam,Vec4(0,0,1,1));

TSound sound=LoadSound("tone.wav");
TSource source=CreateSource(sound,SOURCE_LOOP);
PlaySound(source);

float pitch=1.0;

//Main loop
while(!KeyHit(KEY_ESCAPE)) {

    //Control source
    if (KeyHit(KEY_P)) {
        PlaySource(source);
    }

    if (KeyHit(KEY_SPACE)) {
        if (GetSourceState(source)==SOURCE_PLAYING) {
            PauseSource(source);
        }
        else {
            ResumeSource(source);
        }
    }

    //Adjust pitch
    if (KeyDown(KEY_NUMADD)) {
        pitch=pitch+0.001;
        SetSourcePitch(source,pitch);
    }

    if (KeyDown(KEY_NUMSUBTRACT)) {
        pitch=pitch-0.001;
        SetSourcePitch(source,pitch);
    }

    //Update the world
    UpdateWorld();

    //Render the world
    RenderWorld();

    //Swap the back and front buffers
    Flip();
}

exitapp:
return Terminate();
}

```

3D Spatialization

Because our world is 3D, we need sounds to be heard in 3D as well. 3D spatialization gives us auditory clues about where a sound is emanating from. There are three requirements we must fulfill to hear 3D sounds:

- The sound must be in a mono format.
- We must set the source range.
- We must position the source and/or listener.

3D sounds must be in a mono format. 3D spatialization will not work with stereo sounds.

The file "tone.wav" is a stereo sound. Let's load the mono sound file "airduct.wav" instead:

```
TSound sound=LoadSound("airduct.wav");
```

Now we set the source range. Add this code after the source is created and before the main loop:

```
SetSourceRange(source,20);
```

Now let's declare two new variables before the main loop. These will be used to track the source position:

```
float x=0.0;  
float z=0.0;
```

We can set a source's position in 3D space with the SetSourcePosition command. Add this code inside the main loop:

```
if (KeyDown(KEY_RIGHT)) {  
    x=x+0.1;  
    SetSourcePosition(source,Vec3(x,0,z));  
}  
  
if (KeyDown(KEY_LEFT)) {  
    x=x-0.1;  
    SetSourcePosition(source,Vec3(x,0,z));  
}
```

Here is our complete code. When we press the right arrow key, the sound will move to the right, and we will hear it coming from the right speaker. When we press the left arrow key, the sound will move left, and we will hear it coming from the left speaker:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(800,600);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0,"Error","Failed to create world.",0);
        goto exitapp;
    }

    //Create a camera
    TEntity cam=CreateCamera();
    CameraClearColor(cam,Vec4(0,0,1,1));

    TSound sound=LoadSound("airduct.wav");
    TSource source=CreateSource(sound,SOURCE_LOOP);
    SetSourceRange(source,20);
    PlaySound(source);

    float pitch=1.0;
    float x=0.0;
    float z=0.0;

    //Main loop
    while(!KeyHit(KEY_ESCAPE)) {

        //Source controls
        if (KeyHit(KEY_P)) {
            PlaySource(source);
        }

        if (KeyHit(KEY_SPACE)) {
            if (GetSourceState(source)==SOURCE_PLAYING) {
                PauseSource(source);
            }
            else {
                ResumeSource(source);
            }
        }

        //Adjust pitch
        if (KeyDown(KEY_NUMADD)) {
            pitch=pitch+0.001;
            SetSourcePitch(source,pitch);
        }

        //Position the source
        if (KeyDown(KEY_NUMSUBTRACT)) {
            pitch=pitch-0.001;
            SetSourcePitch(source,pitch);
        }
    }
}
```

```

        if (KeyDown(KEY_RIGHT)) {
            x=x+0.1;
            SetSourcePosition(source,Vec3(x,0,z));
        }

        if (KeyDown(KEY_LEFT)) {
            x=x-0.1;
            SetSourcePosition(source,Vec3(x,0,z));
        }

        //Update the world
        UpdateWorld();

        //Render the world
        RenderWorld();

        //Swap the back and front buffers
        Flip();
    }

    exitapp:
    return Terminate();
}

```

We can also set the position the sound is heard from. A listener entity will automatically update the listening position and orientation when it is moved or rotated. Add this code before the main loop:

```

TListener listener=CreateListener();
PositionEntity(listener,Vec3(0,0,-2));

```

Let's move the camera backwards so it is in the same position as the listener. Add this code after the camera is created and before the main loop:

```

PositionEntity(cam,Vec3(0,0,-2));

```

Let's create a visible mesh to give us a better idea of where the sound is coming from. Add this code before the main loop:

```

TMesh mesh=CreateSphere();

```

Find this block of code in the main loop:

```

if (KeyDown(KEY_RIGHT)) {
    x=x+0.1;

```

```

        SetSourcePosition (source, Vec3 (x, 0, z));
    }

    if (KeyDown(KEY_LEFT)) {
        x=x-0.1;
        SetSourcePosition (source, Vec3 (x, 0, z));
    }

```

Replace it with this code, which will add movement along the z axis and position the visible mesh as well:

```

    if (KeyDown(KEY_RIGHT)) {
        x=x+0.1;
        SetSourcePosition (source, Vec3 (x, 0, z));
        PositionEntity (mesh, Vec3 (x, 0, z));
    }

    if (KeyDown(KEY_LEFT)) {
        x=x-0.1;
        SetSourcePosition (source, Vec3 (x, 0, z));
        PositionEntity (mesh, Vec3 (x, 0, z));
    }

    if (KeyDown(KEY_UP)) {
        z=z+0.1;
        SetSourcePosition (source, Vec3 (x, 0, z));
        PositionEntity (mesh, Vec3 (x, 0, z));
    }

    if (KeyDown(KEY_DOWN)) {
        z=z-0.1;
        SetSourcePosition (source, Vec3 (x, 0, z));
        PositionEntity (mesh, Vec3 (x, 0, z));
    }

```

Here is our final code. When we move the sphere around, it should be apparent the sound is coming from there object's position. The source even sounds more muffled when it is behind the listener:

```

#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics (800, 600);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0, "Error", "Failed to create world.", 0);
        goto exitapp;
    }

    //Create a camera

```

```

TEntity cam=CreateCamera();
CameraClearColor(cam,Vec4(0,0,1,1));
PositionEntity(cam,Vec3(0,0,-2));

TSound sound=LoadSound("airduct.wav");
TSource source=CreateSource(sound,SOURCE_LOOP);
SetSourceRange(source,20);
PlaySound(source);

//Create a listener
TListener listener=CreateListener();
PositionEntity(listener,Vec3(0,0,-2));

//Create a visible mesh
TMesh mesh=CreateSphere();

float pitch=1.0;
float x=0.0;
float z=0.0;

//Main loop
while(!KeyHit(KEY_ESCAPE)) {

    //Source controls
    if (KeyHit(KEY_P)) {
        PlaySource(source);
    }

    if (KeyHit(KEY_SPACE)) {
        if (GetSourceState(source)==SOURCE_PLAYING) {
            PauseSource(source);
        }
        else {
            ResumeSource(source);
        }
    }

    //Adjust pitch
    if (KeyDown(KEY_NUMADD)) {
        pitch=pitch+0.001;
        SetSourcePitch(source,pitch);
    }

    if (KeyDown(KEY_NUMSUBTRACT)) {
        pitch=pitch-0.001;
        SetSourcePitch(source,pitch);
    }

    //Move the source
    if (KeyDown(KEY_RIGHT)) {
        x=x+0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_LEFT)) {
        x=x-0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_UP)) {
        z=z+0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
}

```

```

    }
    if (KeyDown(KEY_DOWN)) {
        z=z-0.1;
        SetSourcePosition (source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }

    //Update the world
    UpdateWorld();

    //Render the world
    RenderWorld();

    //Swap the back and front buffers
    Flip();
}

exitapp:
return Terminate();
}

```

EAX Effects

Leadwerks Engine provides support for hardware EAX effects. This allows the programmer to add echo and reverb to a sound, to simulate the acoustic environment of a tiled bathroom or a spooky cave.

Not all sound cards support EAX, so the first thing we want to do is make sure it can be used on the user's computer. Add this code before the source is created:

```

if (!EAXSupported()) {
    MessageBoxA(0,"Error","EAX is not supported on your computer.",0);
}

```

We can enable an EAX effect with one command. Add this code in the main loop. This will enable EAX effects when the F key is pressed:

```

if (KeyDown(KEY_F)) {
    EAXEffect(EAX_PSYCHOTIC);
}
else {
    EAXEffect(EAX_NORMAL);
}

```

Finally, let's load a more percussive sound that will give a better echo effect:

```

TSound sound=LoadSound("airduct.wav");

```

Here is the complete source code that will play a sound with EAX effects when the F key is pressed:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(800,600);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0,"Error","Failed to create world.",0);
        goto exitapp;
    }

    //Create a camera
    TEntity cam=CreateCamera();
    CameraClearColor(cam,Vec4(0,0,1,1));
    PositionEntity(cam,Vec3(0,0,-2));

    if (!EAXSupported()) {
        MessageBoxA(0,"Error","EAX is not supported on your computer.",0);
    }

    TSound sound=LoadSound("switch.wav");
    TSource source=CreateSource(sound,SOURCE_LOOP);
    SetSourceRange(source,20);
    PlaySound(source);

    //Create a listener
    TListener listener=CreateListener();
    PositionEntity(listener,Vec3(0,0,-2));

    //Create a visible mesh
    TMesh mesh=CreateSphere();

    float pitch=1.0;
    float x=0.0;
    float z=0.0;

    //Main loop
    while(!KeyHit(KEY_ESCAPE)) {

        //EAX controls
        if (KeyDown(KEY_F)) {
            EAXEffect(EAX_PSYCHOTIC);
        }
        else {
            EAXEffect(EAX_NORMAL);
        }

        //Source controls
        if (KeyHit(KEY_P)) {
            PlaySource(source);
        }

        if (KeyHit(KEY_SPACE)) {
            if (GetSourceState(source)==SOURCE_PLAYING) {
                PauseSource(source);
            }
        }
    }
}
```

```

        }
        else {
            ResumeSource(source);
        }
    }

    //Adjust pitch
    if (KeyDown(KEY_NUMADD)) {
        pitch=pitch+0.001;
        SetSourcePitch(source,pitch);
    }

    if (KeyDown(KEY_NUMSUBTRACT)) {
        pitch=pitch-0.001;
        SetSourcePitch(source,pitch);
    }

    //Move the source
    if (KeyDown(KEY_RIGHT)) {
        x=x+0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_LEFT)) {
        x=x-0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_UP)) {
        z=z+0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_DOWN)) {
        z=z-0.1;
        SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }

    //Update the world
    UpdateWorld();

    //Render the world
    RenderWorld();

    //Swap the back and front buffers
    Flip();
}

exitapp:
return Terminate();
}

```

Sound Emission

We can emit sounds from entities, and let the entity automatically manage the source positioning. This is achieved with the EmitSound function as follows:

```
EmitSound(entity, sound);
```

Here is our previous code modified to use EmitSound instead of creating a sound source:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(800,600);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0,"Error","Failed to create world.",0);
        goto exitapp;
    }

    //Create a camera
    TEntity cam=CreateCamera();
    CameraClearColor(cam,Vec4(0,0,1,1));
    PositionEntity(cam,Vec3(0,0,-2));

    if (!EAXSupported()) {
        MessageBoxA(0,"Error","EAX is not supported on your computer.",0);
    }

    TSound sound=LoadSound("switch.wav");
    //TSource source=CreateSource(sound,SOURCE_LOOP);
    //SetSourceRange(source,20);
    //PlaySound(source);

    //Create a listener
    TListener listener=CreateListener();
    PositionEntity(listener,Vec3(0,0,-2));

    //Create a visible mesh
    TMesh mesh=CreateSphere();
    EmitSound(mesh,sound,20,1.0,SOURCE_LOOP);

    float pitch=1.0;
    float x=0.0;
    float z=0.0;

    //Main loop
    while(!KeyHit(KEY_ESCAPE)) {

        //EAX controls
        if (KeyDown(KEY_F)) {
            EAXEffect(EAX_PSYCHOTIC);
        }
        else {
            EAXEffect(EAX_NORMAL);
        }
    }
}
```

```

    /*
    //Source controls
    if (KeyHit(KEY_P)) {
        PlaySource(source);
    }

    if (KeyHit(KEY_SPACE)) {
        if (GetSourceState(source)==SOURCE_PLAYING) {
            PauseSource(source);
        }
        else {
            ResumeSource(source);
        }
    }

    //Adjust pitch
    if (KeyDown(KEY_NUMADD)) {
        pitch=pitch+0.001;
        SetSourcePitch(source,pitch);
    }

    if (KeyDown(KEY_NUMSUBTRACT)) {
        pitch=pitch-0.001;
        SetSourcePitch(source,pitch);
    }
    */

    //Move the source
    if (KeyDown(KEY_RIGHT)) {
        x=x+0.1;
        //SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_LEFT)) {
        x=x-0.1;
        //SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_UP)) {
        z=z+0.1;
        //SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }
    if (KeyDown(KEY_DOWN)) {
        z=z-0.1;
        //SetSourcePosition(source,Vec3(x,0,z));
        PositionEntity(mesh,Vec3(x,0,z));
    }

    //Update the world
    UpdateWorld();

    //Render the world
    RenderWorld();

    //Swap the back and front buffers
    Flip();
}

exitapp:
return Terminate();
}

```

Sound Management

Sound cards only have a limited number of channels. To keep channels available, you must free sources when you are done with them. Freeing a source does not stop the source from playing, it only tells the engine that it is okay to delete the source once it stops playing. Therefore, looped sources must be stopped before they are freed.

You must free sources when you are done with them, or your sound card will run out of channels.

The following code demonstrates an error that can occur if sources are not freed. Keep pressing the P key. Eventually the program will stop playing the sound when your sound card runs out of channels:

```
#include "engine.h"

int main(int argc, char** argv)
{
    Initialize();

    //Create a graphics context
    Graphics(640,480);

    //Create a world
    if (!CreateWorld()) {
        MessageBoxA(0,"Error","Failed to create world.",0);
        goto exitapp;
    }

    //Create a camera
    TEntity cam=CreateCamera();
    CameraClearColor(cam,Vec4(0,0,1,1));

    TSound sound=LoadSound("tone.wav");
    TSource source;

    //Main loop
    while(!KeyHit(KEY_ESCAPE)) {

        //Source controls
        if (KeyHit(KEY_P)) {
            source=CreateSource(sound);
            PlaySource(source);
        }

        //Update the world
        UpdateWorld();

        //Render the world
        RenderWorld();

        //Swap the back and front buffers
        Flip();
    }

    exitapp:
}
```

```
    return Terminate();  
}
```

We can fix this by freeing the source when we are done with it. Remember, freeing a source does not stop it from playing, it only tags it as “deletable” once it is no longer playing:

```
//Source controls  
if (KeyHit(KEY_P)) {  
    source=CreateSource(sound);  
    PlaySource(source);  
    FreeSource(source);  
}
```

An entity that emits a sound will automatically stop and free all sources it is managing when it is freed.