Chapter 7

Sounds Good

In this Chapter I'm going to be showing you how to incorporate Sounds and Music into your applications. Sounds in ShiVa are set up using Sounds that have been imported into ShiVa in the SoundBank Editor. Music is simply imported, and called from Script.

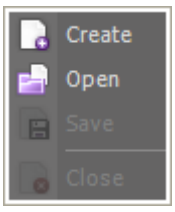
The SoundBank Editor module allows you to edit the contents of a SoundBank. A SoundBank is a set of short resources called 'Sounds'. The Sounds must be quite short (usually less than 10 seconds), and are used to create event sound effects.

In the SoundBank, Sounds are indexed. This indexation allows you to keep all SoundBanks of the same type in a easily manageable format. For example, all SoundBanks attached to a character. In this instance, it is useful to place the 'walk' Sound at index 0, 'run' at index 1, 'jump' at index 2, etc...

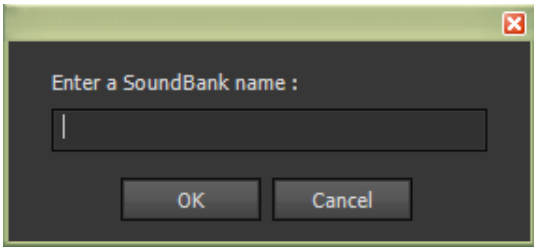
You also can arrange Sounds by modifying their index. So, you could prepare a character with 2 or 3 'walk' Sounds and leave spaces free for another character's Sounds.

The SoundBank Editor looks like this:

To create a new SoundBank, click on "SoundBank" and the following menu will appear:

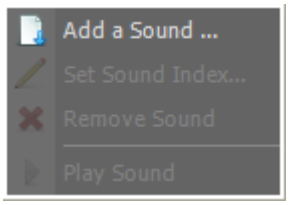


By clicking on "Create", a dialog will appear asking for the name of the new SoundBank:

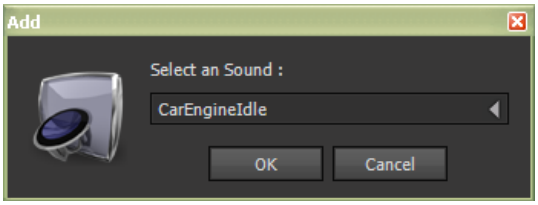


Enter the name of your new SoundBank, and click on "OK". Your new (empty) SoundBank will now appear in the SoundBank Editor.

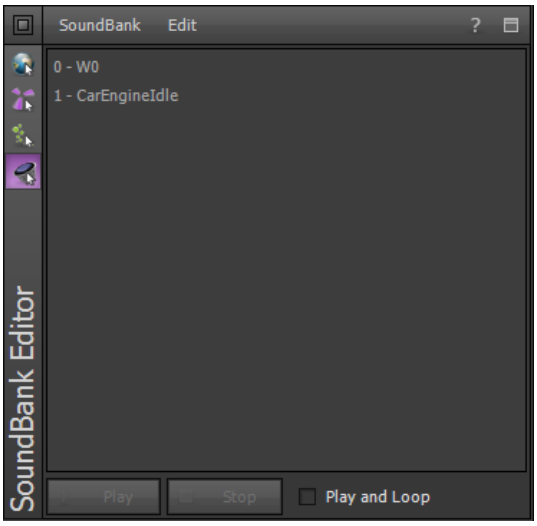
To add Sounds to your SoundBank, click on "Edit" and the following menu will appear:



Click on "Add a Sound ...", and all Sounds available in your Project will be displayed in a drop-down list in the dialog box that appears:

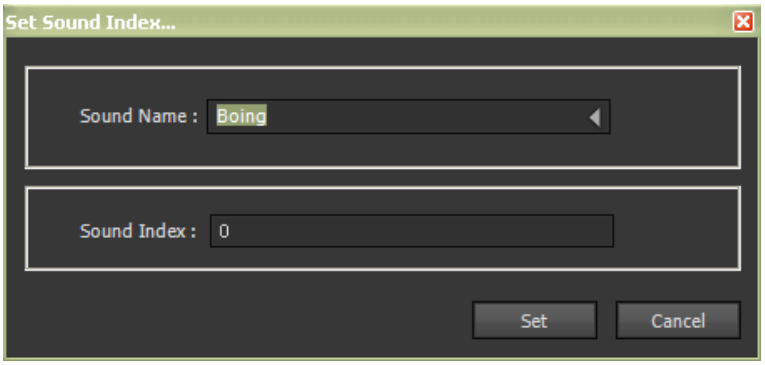


So, you simply select the Sound to add to the SoundBank, and click on "OK". The added Sound will now appear in the SoundBank Editor. Note that each Sound is given an Index number before the name of the Sound, as shown below:



In my case, I have two Sounds, "W0" at Index "0" and "CarEngineIdle" at Index "1".

To change the Index numbers of the Sounds, you can click on the "Edit" menu. In the menu that appears, select "Set Sound Index...", and the following dialog will appear:



In this dialog, you can select the Sound from the drop-down list, and set the Index number directly in the "Sound Index" input box.

NOTE: if the chosen Index is already being used, a message will appear asking if the two Sounds should be swapped.

NOTE: a Sound can also be dragged and dropped directly into the required position.

The "Remove Sound" option in the "Edit" menu allows the deletion of the selected Sound from the SoundBank, and the "Play Sound" option allows the playing, or stopping, of the selected Sound.
NOTE: the "Play Sound" option will change to "Stop Sound" if a Sound is currently playing.

The two buttons at the bottom of the Sound Editor ("Play" and "Stop") allow the playing, and stopping, of the sound, as if the "Play Sound / Stop Sound" menu option was selected.

If the "Play and Loop" option is checked, then the sound will play in a continuous loop until it is stopped.

Now that you've set up your SoundBank, I'll show you how to create a little app that uses both Music and your new SoundBank.

OK, for Sounds, there is a demo available (called "PonctualSounds") in the standard ShiVa demos ("Installation_Samples.zip").

So, you will need to create a new project in ShiVa.

This demo consists of the following files (in addition to those automatically created by ShiVa when you create a new project):

Games	-	PonctualSounds
Models	-	StaticPointLight PunctualSound_Ball Box (Shape) Box (Shape)
AIModels	-	PunctualSound_Ball PunctualSound_Main

Materials	-	cornell_box_Material000 cornell_box_Material001 cornell_box_Material002 cornell_box_Material003 cornell_box_Material004 PunctualSound_Ball PunctualSound_Ground
Meshes	-	cornell_box_Mesh000 cornell_box_Mesh001 cornell_box_Mesh002 cornell_box_Mesh003 PunctualSound_Ball
Scripts	-	Fire_Main_Function_loadScene Fire_Main_Handler_onInit
SoundBanks	-	Ball
Sounds	-	Boing
Scenes	-	PunctualSound

Again, the ones highlighted in Red will be the files that I'll be dissecting here. Since this Chapter is about Sound, we will start with the SoundBank:

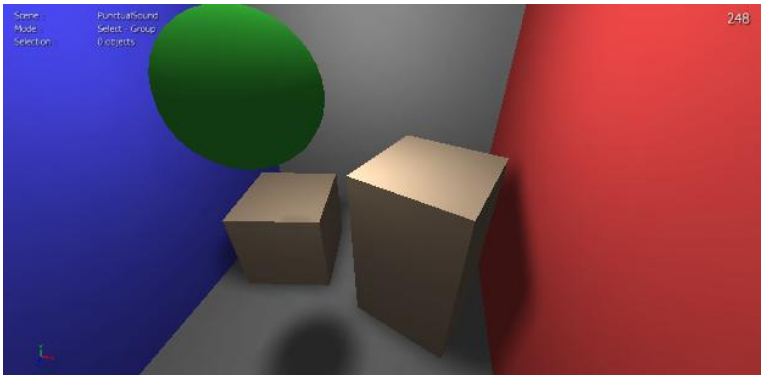
So, open up the SoundBank Editor, and we'll get started.

If you click on the "SoundBank" menu, and then "Open", you can select the "Ball" SoundBank, and you should see this:

Not very exciting, but you can see one Sound ("Boing") at index 0. If you click on the Sound, you can then click on the "Play" button, and the Sound will be played.

Well, that's it for the SoundBank Editor! The Sound was added using the "Edit" menu, selecting "Add a Sound ...", and then selecting the required Sound from the drop-down box.

OK, I'll move on now to the Scene for this demo, so fire up the Scene in the Scene Viewer, and I'll get started:



Fairly standard ShiVa demo stuff, we have a ball, two boxes, and the cornell box, which has been set as a "Collider".

If you now right-click on the "ball", and navigate to the "Controllers" sub-menu, you will see an option called "Sound". If you select this, you will see that the options are "Edit SoundBank", and "Remove SoundBank". This shows that there is a SoundBank attached to this Model (if there wasn't, you would have the "Add" option in place of the "Edit"). Note that you can only have one SoundBank attached to a Model.

OK, so you now have a Scene, with a Model that has a SoundBank attached. What's next? Scripting again!

This demo consists of the following Scripts:

PunctualSound_Ball

This AIModel consists of the following:

Variables

nAverageSpeed - Number - Initial Value – 0.000

Functions

setupDynamics ( )

Handlers

onEnterFrame ( )
onInit ( )

setupDynamics ( )

```
local o = this.getObject ( )

if ( dynamics.createSphereBody ( o, 1 ) )
then
    dynamics.enableDynamics ( o, true )
    dynamics.enableCollisions (o, true )
    dynamics.enableGravity (o, true )
    dynamics.setBounce ( o, 1 )
end
```

All that these lines do, is create a "Dynamics" Sphere around the Object, and then set some Parameters for the Dynamics. The only thing I'll point out here is the "setBounce" line. This sets the bounciness of the Dynamics, by setting the bouncing coefficient (in this case to "1"). Note that the lower the value is for this Parameter, the less the bounce. Just for fun, try setting it to "2" and see what happens!

onEnterFrame ( )

```
local o = this.getObject ( )

local avgSpeed = 0.5 * ( this.nAverageSpeed ( ) + dynamics.getLinearSpeed ( o ) )

this.nAverageSpeed ( avgSpeed )
```

OK, up to here, all we are doing is calculating the average speed of the ball. This is done by taking half of the sum of the previous average ("nAverageSpeed ( )" and the current linear speed of the Dynamics Object ("dynamics.getLinearSpeed (o)").

```
if ( dynamics.getLastCollisionTime ( o ) < 0.1 and avgSpeed > 2 )
then
    sound.play ( o, 0, 1, false, 0.5 )
end
```

These last lines check to see if the last "Collision" time is less than "0.1" and the current average speed, as calculated above, is greater than "2". If so, the Sound is played using the following Parameters:

o	-	the Object
0	-	the SoundBank index of the Sound to be played
1	-	the Volume (between 0 and 1, with 1 being full volume)
false	-	no repeat (i.e. play the Sound once only)
0.5	-	the priority of the Sound (since this is the only Sound playing, this has no effect)

onInit ( )

This Script has one line, which calls the "setupDynamics" Function:

```
this.setupDynamics ( )
```

PunctualSound_Main

This AIModel consists of the following:

Functions

loadScene ( )

Handlers

onInit ( )

loadScene ( )

This Script consists of just one line:

application.setCurrentUserScene ("PunctualSound")

All this line does is set the current Scene for the User to the "PunctualSound" Scene.

onInit ( )

this.loadScene ( )

All this does is call the "loadScene" Function above.

That's it for this demo! Note that I'll be covering Music in Chapter 11 when I explain the Ambience Editor.

Well, that's it for another Chapter, next we'll be delving into the colourful world of Materials.