

Torque Game Builder – Audio Tutorial

Introduction

Sound effects are a very important component of game immersion. As such, Torque Game Builder (TGB) provides users a robust sound management system for assigning and playing sound effects in-game. In this tutorial we will show you how to create your own sound descriptions and profiles, load your sound into your game, and show you some of the ways you can manipulate sound in TGB. For more reference on this subject, please refer to the *TGB Reference* document that is provided with Torque Game Builder.

Setting Up Our Sound Files

Our first order of business is to get some sound files to mess around with. In your `games/scrollerDemo/data/audio` folder you will see all of the sound files that were used in the demo. For the purposes of our tutorial, we are going to grab two of them: `Abient06` and `Fire17`. Copy these files and paste them into your `games/TGB/data/audio` folder. As you might have guessed, we are going to use our *TGB* stock project to test our sound in.

In order for sounds to be accessed by TGB, they must have an audio description and an audio profile. An audio description sets the properties of the sound and the profile sets the name of the sound, the description to use, and the actual audio file to be referenced. For this tutorial, we are going to create two audio descriptions, one for looping sound and one for non-looping sound.

Navigate to your `games/TGB/gameScripts` folder and create a new `.cs` file named “`audioDatablocks.cs`”. Audio descriptions and profiles can be accessed anywhere in script; however, for the sake of organization, we are creating our audio code in a separate file. First, we need to set up our audio descriptions. Add the following code to your `audioDatablocks.cs` file:

```
new AudioDescription(AudioNonLooping)
{
    volume = 1.0;
    isLooping= false;
    is3D = false;
    type = $GuiAudioType;
};

new AudioDescription(AudioLooping)
{
    volume = 1.0;
    isLooping= true;
    is3D = false;
    type = $GuiAudioType;
};
```

Torque Game Builder – Audio Tutorial

This code simply sets up two new audio descriptions, one for looping sounds, and one for non-looping sounds. Next, we will need two audio profiles, one for each of the sounds we are using in this tutorial. Add this code below your two audio descriptions:

```
// -----  
// Background Loop.  
// -----  
new AudioProfile(backgroundAudio)  
{  
    filename = "~/data/audio/Ambient06.wav";  
    description = "AudioLooping";  
    preload = true;  
};  
  
// -----  
// Fire Sound.  
// -----  
new AudioProfile(fireAudio)  
{  
    filename = "~/data/audio/Fire17.wav";  
    description = "AudioNonLooping";  
    preload = true;  
};
```

As you can see, each of our audio profiles have specified one of the two audio descriptions that we created earlier. They also reference the path to the sounds each will use. That should be enough code to get our audio working – make sure to exec your `audioDatablocks.cs` in `game.cs` so that TGB will run the code.

Testing Our Sounds

Now, we are going to test our sound files and show you some of the ways Torque Game Builder can manipulate sounds. Since we defined our audio descriptions and profiles, we are given the use of a few commands that make manipulating our sound files very easy. Open up Torque Game Builder and load your *TGB* project by choosing *Open Project* from the file menu. When you get your project loaded, immediately hit the play button. NOTE – if you loaded an unsaved level, you will have to save it before you can play.

Once you are inside the game, hit the tilde (~) key to bring up your console. The first thing we are going to show you how to do is play a sound. Inside the console, enter the following line:

```
alxPlay(fireAudio);
```

When you hit enter, you should hear the firing sound play once and then stop. You will notice that we passed the `alxPlay` function the name of our audio profile. The `alxPlay` function returns a handle; however, since the sound plays once then disappears, it was not necessary

Torque Game Builder – Audio Tutorial

to save that handle. However, that is not always the case, as we shall now demonstrate.

Next, type the following line into your console:

```
$sound = alxPlay(backgroundAudio);
```

When you hit enter, you should hear the ambient audio track playing. Since this audio profile uses the looping description, it should continue playing over and over. You will notice that this time around we saved our audio handle in a global variable \$sound. Next, type this into the console:

```
alxStop($sound);
```

Your sound should stop immediately. You should note that the alxStop command only accepts the sound handle created by alxPlay as a parameter. In addition to stopping single sounds, there is a function that will stop all sounds currently being played. Type the following command into your console four or five times, hitting enter in between commands:

```
alxPlay(backgroundAudio);
```

After you call that command 2 or 3 times, you should hear something pretty horrible: multiple sound tracks playing out of sync. Type the following command into your console:

```
alxStopAll();
```

This should stop all sounds being played immediately.

Conclusion

This tutorial has stepped you through creating audio descriptions and profiles. In addition, you have been shown several of the commands you can use to manipulate sounds in your scene. For further information, please refer to the *TGB Reference* document that is included with Torque Game Builder.