Notes from Camille Goudeseune's Sound Lecture

William Baker

August 9, 2004

Abstract

Camille covered the following topics:

- VSS History and Introduction
- Sound in Syzygy
- Syzygy Code Example

1 VSS History and Introduction

VSS is a system designed and built for the CAVE between 1994 and 1995. VSS's stregths were that it provided the ability to do multi-channel audio and real-time audio computation (one example of such real-time audio is in the Doppler Glom program in the CAVE). The sound control in Syzygy doesn't provide as powerful a funtionality as VSS, but as will be seen in the next section, sound in Syzygy is much easier to control and implement.

2 Sound in Syzygy

2.1 The Scene-Graph Model

Sound in Syzygy is based on the scene-graph model. This system is near identical to the distributed scene-graph model for graphics, except that sounds are attached to nodes instead of graphics objects. For those unfamiliar with the scene-graph framework, what a programmer does is specifya location in the scene to place a graphic or sound (in this case, sounds). However, one must keep in mind that the scene-graph for graphics and the scene-graph for sounds are seperate. There is not necessarily a tight relationship between sound and visuals (one might want background sounds unrelated to visuals) and hence there are seperate scene-graphs.

2.2 Types of Sounds in Syzygy

2.2.1 Sound File Types

Syzygy sound supports the following sound file types:

- *.wav uncompressed format
- *.mp3 MPEG-3 compression format

2.2.2 Triggered Sounds

Triggered sounds are sounds that are a response to an event in the program. These can be the result of button-presses, collisions, or any other type of event that a programmer might want to specify. Code for specifying a triggered event is included in the code example section of the notes.

2.2.3 Looping Sounds

Looping sounds, as the name implies, play more than once after they are started (i.e. they loop until a command is issued to stop them from playing).

3 Syzygy Sound Code Examples

3.1 dsLoop()

The Syzygy functions dsLoop() will be the primary way that sounds are added, started, and stopped within Syzygy. The dsLoop() function can be used in the three different ways presented below. It is important to note that in the following 3 calls with dsLoop(), all of the parameters are not supplied, only the parameter that affects what dsLoop() does when it's called.

3.1.1 dsLoop(1)

Calling dsLoop() with a parameter of "1" will start a sound loop using the user-defined sound file. This sound will continue until a subsequent call to dsLoop with a value of "0" is made.

3.1.2 dsLoop(0)

Calling dsLoop() with a "0" parameter serves two purposes. First, this type of call to dsLoop can be used to stop a looping sound that was previously started in the program with a call of dsLoop(1). The second purpose that dsLoop(0) serves is to reset a trigger sound. Once a triggered sound effect is called, it cannot be called again until dsLoop(0) is called. One common technique is to call dsLoop(0) immediately following the dsLoop(-1) that starts a triggered sound.

3.1.3 dsLoop(-1)

This final dsLoop() call with a value of "-1" is used with triggered sounds. dsLoop(-1) will start playing a sound and then stop with the sound file reaches its end. As mentioned above, to call the triggered sound again, dsLoop(0) will

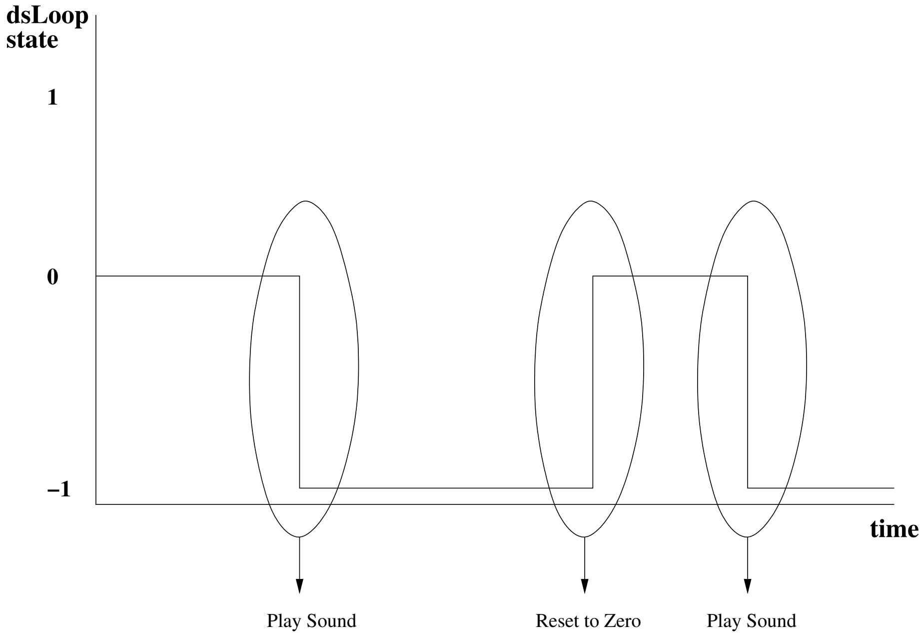

Figure 1: How the state of dsLoop() its transitions between states relate to triggered sounds being played.

need to be called to "reset" the sound. One can imagine this process as a state diagram. The "0" state is when a sound is ready to be played, and the "-1" state represents a sound that has been played and should not be played again. If a sound is in the "0" state and a call of dsLoop(0) occurs nothing happens. If there is a call of dsLoop(-1), then the current state becomes "-1" and the sound is played. While in the "-1" state, if calls of dsLoop(-1) are encountered nothing happens, and a call of dsLoop(0) changes the current state from "-1" to "0". See Figure 1 for one representation of this process and see Figure 2 for the state diagram.

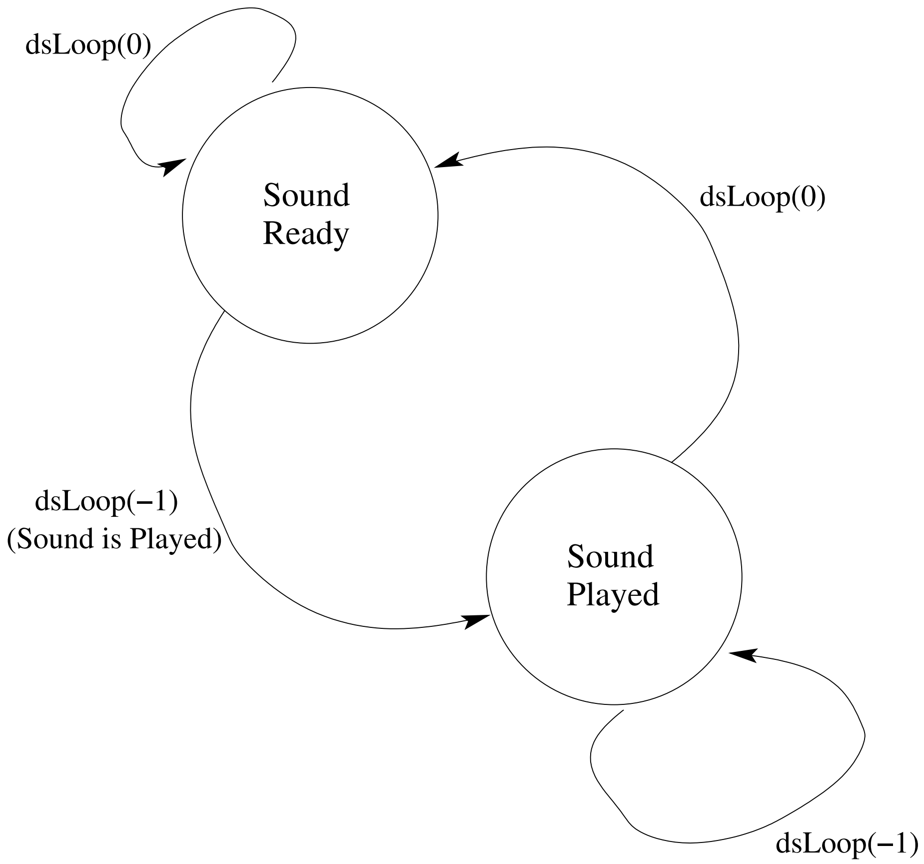

Figure 2: State diagram for triggered sounds using dsLoop().

3.2 Code example using dsLoop

```
...
#Note below that "ds..." in "dsTransform" refers to sound
sID=dsTransform('world',fw.getNavNodeName(),arMatrix4())

#dsLoop(name, what to attach to, file name, mode (see above explanation), max
#       amplitude between 0 and 1, location sound comes from)
dsLoop('unchanging','world','cosmos.mp3',1,1.0,arVector3(0,0,0))



#When doing sound events, first set a variable equal to dsLoop(...)
#Then later call dsLoop(variable name, -1 or 0, amplitude, vector)
```