

Task Proposal

Allen Pestaluky

Lecturer: Dr. Chris Joslin
Feb. 27, 2008
Term Project Part A, IMD3004

1 Introduction

This task proposal document outlines the objectives and basic design elements of a Virtual Studio Technology (VST) effect plug-in that will be completed for *IMD 3004: Human Computer Interaction & Design*. Mock-ups and diagrams will be used to help explain the design of certain components of the VST plug-in. This document contains two main elements:

1.1 [Critical Elements]

Critically important elements of the VST effect plug-in will be written in **black text**. These elements are fundamental to the project and are intended to be fully functional in the final submission of the project.

1.2 [Non-Critical Elements]

Elements that are not essential to the full completion the VST effect plug-in will be written in **blue text**. Though these elements **are** intended to be fully functional in the final submission of the project, they may be removed if time restrictions prevent the completion of the plug-in.

2 Virtual Studio Technology

The Virtual Studio Technology (VST) interface developed by Steinberg is provided as a free software development kit (SDK). A VST plug-in created with this SDK can be integrated into any VST compatible software environment such as Steinberg's [Cubase](#), Adobe's Audition, or other free software such as [Wavosaur](#). It is possible to create VST effects and VST instruments. Effects can be applied to any audio channel, while instruments can be applied to any MIDI channel. This goal of this project is to develop a VST *effect* plug-in.

2.1 Developing and Using VST Effect Plug-Ins

There are two components to a VST effect plug-in:

2.1.1 Audio Processing

The audio processing component is all that is required to create a VST effect plug-in. This component is what performs changes to the audio stream. Sample projects, provided with the VST SDK 2.4, have been successfully built and tested at the present time. These sample projects made use of only the audio processing component of the VST effect plug-in.

2.1.2 User Interface

"All user-interface issues are entirely separated from the audio processing issues. At its simplest there is an option where you can avoid providing a user interface at all. In this case the host requests character strings from the Plug-In representing each of the parameters. The host can use the separate ASCII strings for the value, the label, and the units of the parameters to construct its own user interface." [VST SDK 2.4 Documentation]

A VST Graphical User Interface (GUI) SDK is provided as a [free SourceForge Project](#). This SDK will be used to develop a GUI for this project.

*Note: When using some VST compatible software, the VST GUI will **not** be displayed even when it is present in the plug-in. Instead, all controls will be presented through the host application's own interface. The free Wavosaur software behaves like this.*

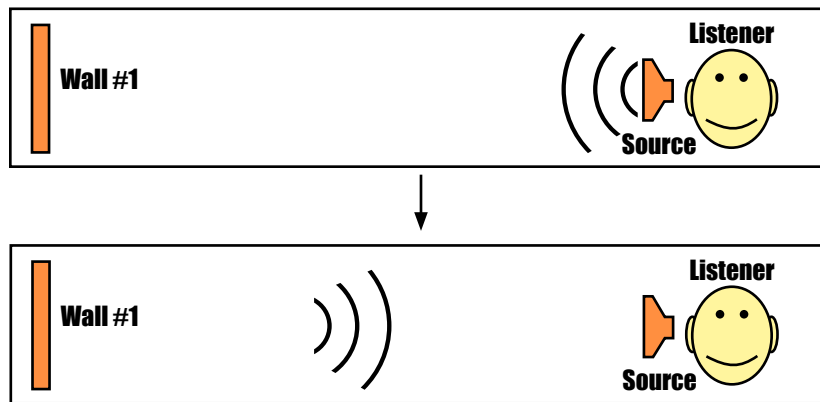
3 Global Objective

A VST audio effect plug-in will be created that will simulate reflections off of a surface as well as replicate human auditory spatial perception cues.

4 Sub-Objectives

4.1 Wall Simulation

The VST effect plug-in should simulate the effect of sound being reflected off a wall at a perpendicular angle and returning to it's source location.

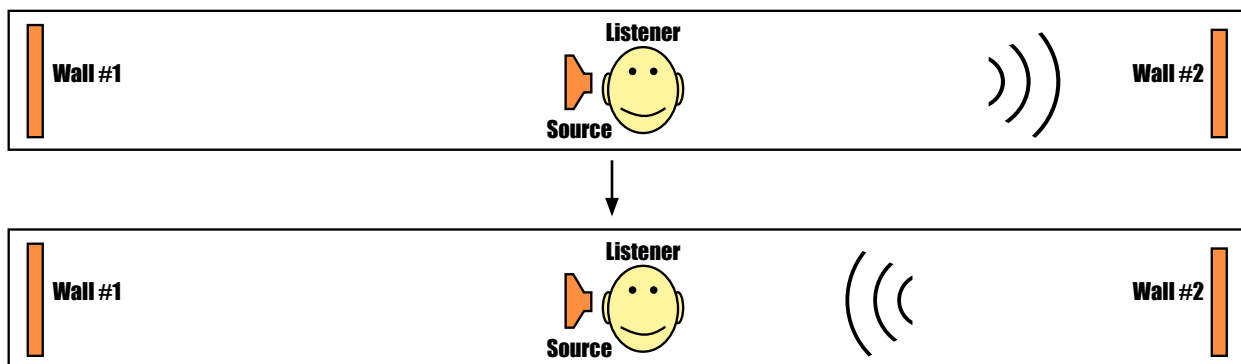


4.2 Spatial Perception Cues

Spatial perception cues should be used to give the impression of the listener facing a specific direction relative to the wall that is reflecting the sound.

4.3 Second Wall

A second wall should be used to reflect the sound that has been reflected off the first wall back to the listener and the first wall.



5 Specifications

5.1 Wall Simulation

5.1.1 Distance (Delay)

A delay will be applied to the audio stream to simulate the time it takes for a sound wave to travel a given distance.

5.1.2 Refraction (Pitch Shift)

A simplified one-way refraction will be applied in the form of a pitch shift to the sound wave to simulate passing through a change in medium. This is not possible in real life when reflecting off a surface, because the sound wave will return to its original medium before returning to the listener. Though this does not represent a realistic reflection, the option will be included in the VST plug-in.

5.1.3 Reflectivity (Gain)

The reflectivity of the wall surface will be controlled by a “gain” value.

5.1.4 Frequency Specific Reflectivity (Gain EQ)

A simple equalizer will be included to give control over how strongly specific frequencies are reflected by the wall.

5.2 Spatial Perception Cues

Control of the listener’s angle relative to the reflecting wall will be provided. This angle will be used by the following processes just before the sound wave is given to the final output:

5.2.1 Stereo Balance (for use on speakers)

The angle of the listener will be used to adjust the balance of the audio stream between the left and right channels.

5.2.1 ITD and IID (for use on headphones)

Inter-Aural Time Difference (ITD) and Inter-Aural Intensity Difference (IID) will be applied for use on headphones that do not naturally reproduce this effect. A small delay between the left and right audio channel will be applied to simulate ITD. A drastic gain reduction in one channel will be applied to simulate IID. One of these two elements will be used at any one time depending on the angle given. Exact values relative to the angle will be researched and applied to the plug-in.

5.2.2 Head Related Frequency Response (HRFR)

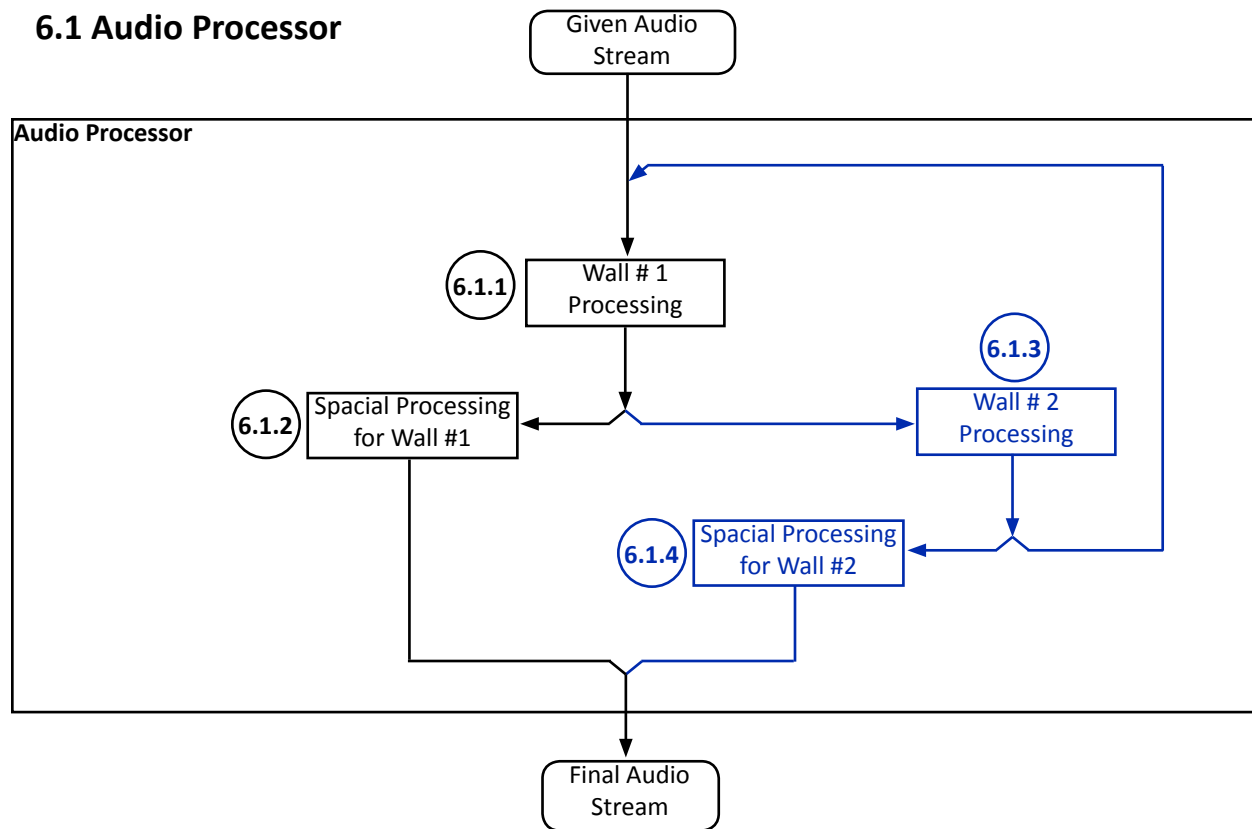
HRFR will be applied to the audio stream to simulate natural spatial perception cues. The angle will be adjusted to the nearest angle with HRFR data available if no algorithm can be generated in the scope of this project.

5.3 Second Wall

A second wall will be positioned directly behind the other wall and will have all of the same control abilities as the first wall. After reflecting off the second wall, the audio stream will be given to the listener’s spatial processing as well as given to the first wall to be reflected again.

6 Design

6.1 Audio Processor



6.1.1 Wall #1 Processing

The delay, pitch shift, gain (reflectivity), and equalization (frequency specific reflectivity) will be processed. The result will be split to the spacial processing and wall #2.

6.1.2 Spacial Processing for Wall #1

Depending on the control's values, the HRRF will be processed and then the stereo balance or ITD and IID will be processed with the given angle.

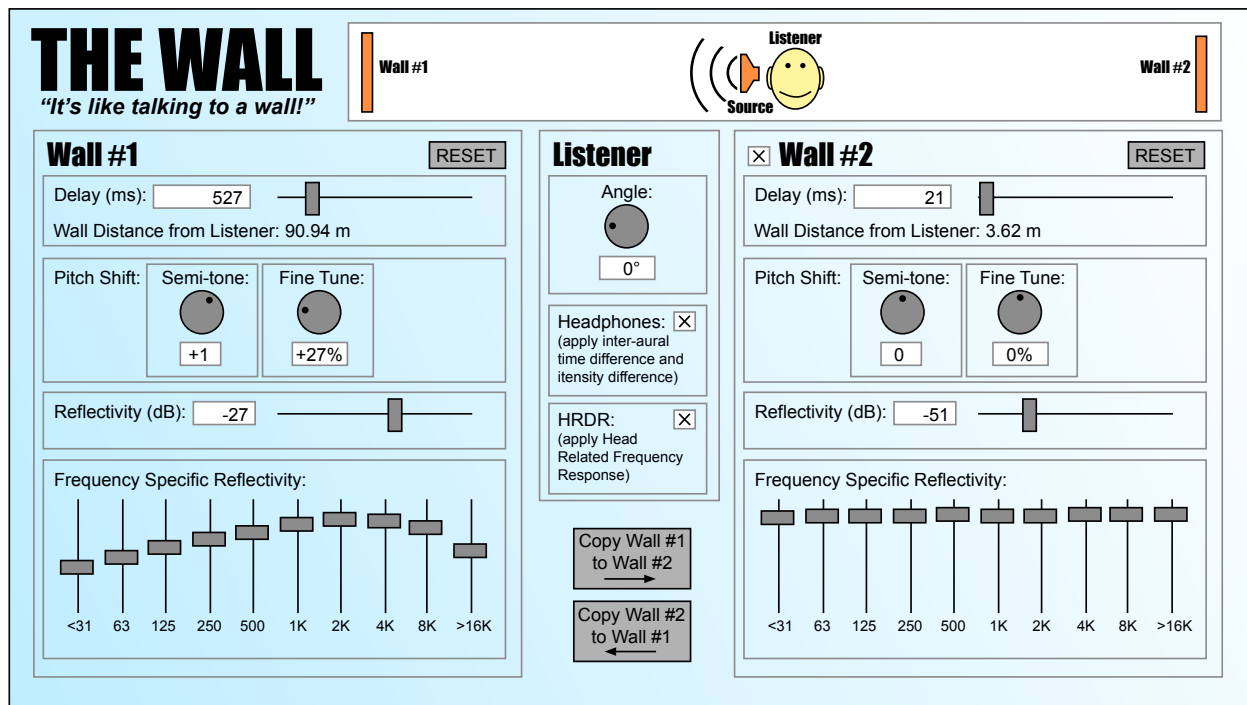
6.1.3 Wall #2 Processing

If Wall #2 is enabled, processing similar to the first wall will be performed with the control values from Wall #2. Depending on the architecture of the SDK, this will most likely be an object of the same class as Wall #1. The result will be mixed with the "Given Audio Stream" that will be fed into "Wall #1 Processing".

6.1.4 Spacial Processing for Wall #2

If Wall #2 is enabled, processing similar to the first wall's spacial processing will be performed with a reverse angle. Depending on the architecture of the SDK, this will most likely be an object of the same class as "Spacial Processing for Wall #1". The result will be mixed with the audio stream from "Spacial Processing for Wall #1" and returned to the host application.

6.2 GUI



6.2.1 GUI Layout and Components

The GUI layout has been kept simple, but compact. In most audio mixing environments, screen space is at a premium, so a design with very little wasted space has been created.

A diagram has been included in the top of the dialogue to visually show the user what the effect is supposed to replicate.

Wall #1 and the Listener are always active, but Wall #2 can be turned on and off by using the checkbox beside “Wall #2”. The controls from Wall #2 will be greyed out and become inactive when Wall #2 is disabled.

The “Wall Distance from Listener” will be given by multiplying the delay (in milliseconds) by the speed of sound (0.345125 m/ms) and dividing by two.

The angle of the listener will be measured in degrees from 180° to -180° relative to Wall #1 where facing Wall #1 is 0°.

Each wall includes a “RESET” button that will reset the controls of that wall to their default values. It is also possible to copy the control values from one wall to the other by using the “Copy Wall” buttons near the middle bottom of the dialogue.

6.2.2 Presets

Presets will be available, but are handled and displayed by the host GUI. The presets will store the values of every control, or parameter, of the interface.